

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



PTO/SB/21 (08-03)

Approved for use through 08/30/2003. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

		Application Number	10/605,560
		Filing Date	10/08/2003
		First Named Inventor	Tun-Hsing Liu
		Art Unit	
		Examiner Name	
Total Number of Pages in This Submission	3	Attorney Docket Number	MTKP0041USA

ENCLOSURES (Check all that apply)

<input checked="" type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/ Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation <input type="checkbox"/> Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance communication to Technology Center (TC) <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input type="checkbox"/> Other Enclosure(s) (please Identify below):
Remarks		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Winston Hsu, Reg. No.: 41,526
Signature	
Date	12/18/2003

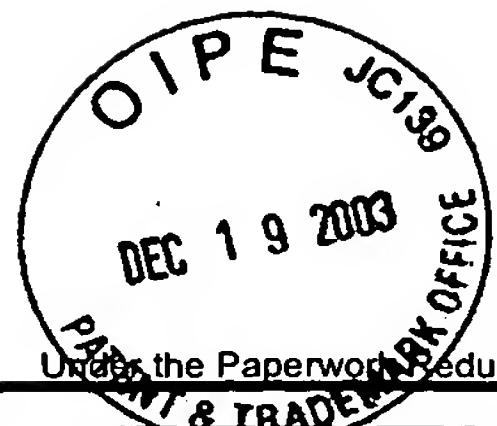
CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name		
Signature	Date	

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PTO/SB/17 (10-03)

Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$0.00)

Complete if Known

Application Number	10/605,560	S
Filing Date	10/08/2003	
First Named Inventor	Tun-Hsing Liu	
Examiner Name		
Art Unit		
Attorney Docket No.	MTKP0041USA	

METHOD OF PAYMENT (check all that apply)

Check Credit card Money Order Other None

Deposit Account:

Deposit Account Number	50-0801
Deposit Account Name	North America International Patent Office

The Director is authorized to: (check all that apply)

Charge fee(s) indicated below Credit any overpayments
 Charge any additional fee(s) or any underpayment of fee(s)
 Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid
1001 770	2001 385	Utility filing fee	
1002 340	2002 170	Design filing fee	
1003 530	2003 265	Plant filing fee	
1004 770	2004 385	Reissue filing fee	
1005 160	2005 80	Provisional filing fee	
SUBTOTAL (1)		(\$0.00)	

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

Total Claims	Independent Claims	Multiple Dependent	Extra Claims	Fee from below	Fee Paid
			-20** =	X	=
			- 3** =	X	=

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description
1202 18	2202 9	Claims in excess of 20
1201 86	2201 43	Independent claims in excess of 3
1203 290	2203 145	Multiple dependent claim, if not paid
1204 86	2204 43	** Reissue independent claims over original patent
1205 18	2205 9	** Reissue claims in excess of 20 and over original patent
SUBTOTAL (2)		(\$0.00)

**or number previously paid, if greater; For Reissues, see above

3. ADDITIONAL FEES

Large Entity	Small Entity	Fee Description	Fee Paid
Fee Code (\$)	Fee Code (\$)		
1051 130	2051 65	Surcharge - late filing fee or oath	
1052 50	2052 25	Surcharge - late provisional filing fee or cover sheet	
1053 130	1053 130	Non-English specification	
1812 2,520	1812 2,520	For filing a request for ex parte reexamination	
1804 920*	1804 920*	Requesting publication of SIR prior to Examiner action	
1805 1,840*	1805 1,840*	Requesting publication of SIR after Examiner action	
1251 110	2251 55	Extension for reply within first month	
1252 420	2252 210	Extension for reply within second month	
1253 950	2253 475	Extension for reply within third month	
1254 1,480	2254 740	Extension for reply within fourth month	
1255 2,010	2255 1,005	Extension for reply within fifth month	
1401 330	2401 165	Notice of Appeal	
1402 330	2402 165	Filing a brief in support of an appeal	
1403 290	2403 145	Request for oral hearing	
1451 1,510	1451 1,510	Petition to institute a public use proceeding	
1452 110	2452 55	Petition to revive - unavoidable	
1453 1,330	2453 665	Petition to revive - unintentional	
1501 1,330	2501 665	Utility issue fee (or reissue)	
1502 480	2502 240	Design issue fee	
1503 640	2503 320	Plant issue fee	
1460 130	1460 130	Petitions to the Commissioner	
1807 50	1807 50	Processing fee under 37 CFR 1.17(q)	
1806 180	1806 180	Submission of Information Disclosure Stmt	
8021 40	8021 40	Recording each patent assignment per property (times number of properties)	
1809 770	2809 385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810 770	2810 385	For each additional invention to be examined (37 CFR 1.129(b))	
1801 770	2801 385	Request for Continued Examination (RCE)	
1802 900	1802 900	Request for expedited examination of a design application	
Other fee (specify)			
*Reduced by Basic Filing Fee Paid		SUBTOTAL (3) (\$0.00)	

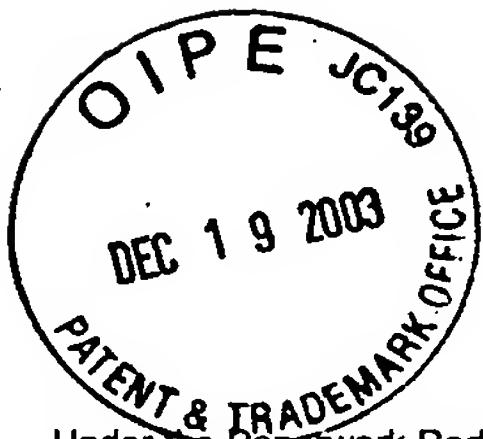
(Complete if applicable)

Name (Print/Type)	Winston Hsu	Registration No. (Attorney/Agent)	41,526	Telephone 886289237350
Signature			Date	12/8/2003

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



PTO/SB/02B (11-00)

Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

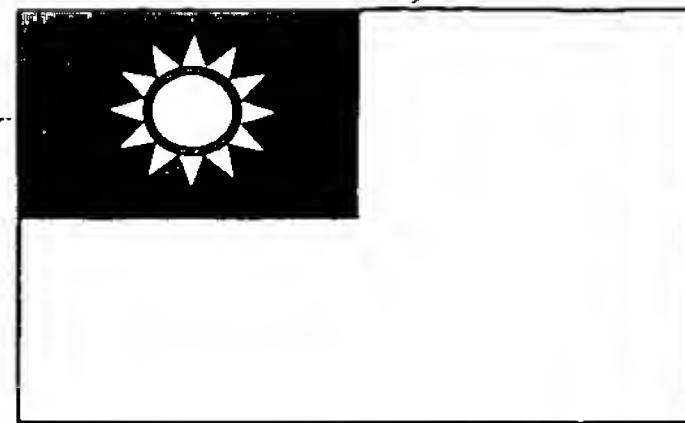
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

DECLARATION -- Supplemental Priority Data Sheet

Additional foreign applications:

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached?
			<input type="checkbox"/>	<input checked="" type="checkbox"/> YES <input type="checkbox"/> NO
092101069	Taiwan R.O.C	01/20/2003	<input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>

Burden Hour Statement: This form is estimated to take 21 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，
其申請資料如下：

This is to certify that annexed is a true copy from the records of this office of the application as originally filed which is identified hereunder:

申 請 日：西元 2003 年 01 月 20 日
Application Date

申 請 案 號：092101069
Application No.

申 請 人：聯發科技股份有限公司
Applicant(s)

局 長
Director General

蔡 緣 生

發文日期：西元 2003 年 3 月 19 日
Issue Date

發文字號：09220270900
Serial No.

申請日期：

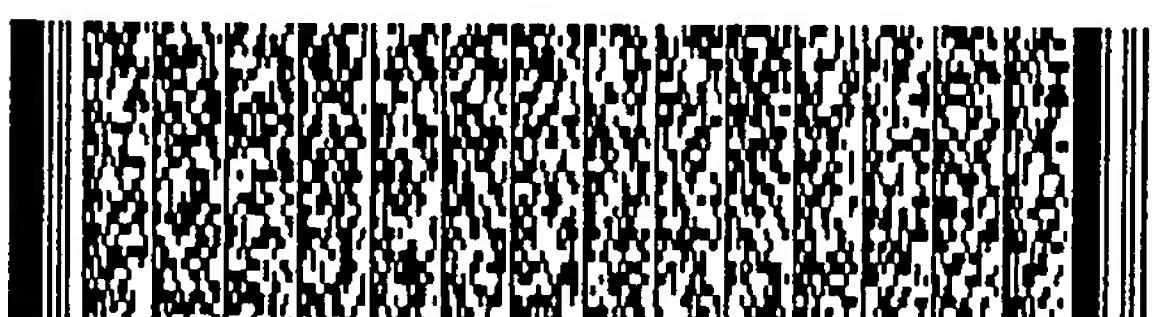
IPC分類

申請案號：

(以上各欄由本局填註)

發明專利說明書

		以檢查韌體程式內容確保韌體更新相容性之韌體更新方法及相關裝置
一、 發明名稱	中 文	
	英 文	Firmware Updating Method And Related Apparatus For Checking Content of Replacing Firmware Before Firmware Updating
二、 發明人 (共2人)	姓 名 (中文)	1. 劉敦行 2. 吳元丁
	姓 名 (英文)	1. Liu, Tun-Hsing 2. Wu, Yuan-Ting
	國 籍 (中英文)	1. 中華民國 TW 2. 中華民國 TW
	住居所 (中 文)	1. 雲林縣斗六市莊敬路三六八號九樓 2. 新竹市民享一街二十二巷二十三號
	住居所 (英 文)	1. 9F, No. 368, Chuang-Ching Rd., Tou-Liu City, Yun-Lin Hsien, Taiwan, R.O.C. 2. No. 23, Lane 22, Ming-Shiang 1st St., Hsin-Chu City, Taiwan, R.O.C.
三、 請人 (共1人)	名稱或 姓 名 (中文)	1. 聯發科技股份有限公司
	名稱或 姓 名 (英文)	1. MediaTek Inc.
	國 籍 (中英文)	1. 中華民國 TW
	住居所 (營業所) (中 文)	1. 新竹市新竹科學工業園區創新一路13號1F (本地址與前向貴局申請者相同)
	住居所 (營業所) (英 文)	1. 1F, No. 13, Innovation Road 1, Science-Based Industrial Park, Hsin-Chu City, Taiwan, R.O.C.
代表人 (中文)	1. 蔡明介	
	代表人 (英文)	1. Tsai, Ming-Kai



四、中文發明摘要（發明名稱：以檢查韌體程式內容確保韌體更新相容性之韌體更新方法及相關裝置）

本發明係提供一種於一電子系統中更新韌體的方法及相關裝置。該電子系統中有一主機及一周邊裝置，該周邊裝置中設有一控制電路及一快閃記憶體，該控制電路係執行一記錄於該快閃記憶體中的第一程式碼以接收該主機之控制指令而操控該周邊裝置的運作。而該方法係在一第二程式碼取代/更新該第一程式碼之前，進行一檢查步驟以檢查該第二程式碼的內容是否符合一預設之內容，以判斷該第二程式碼的相容性。該檢查步驟可由該主機或/及該控制電路進行，該檢查步驟可檢查該第二程式碼中記錄的部分常數值、字串是否符合預設值，且檢查該第二程式碼中位於預設位址的指令、資料是否符合預設之指令、資料，以判斷該第二程式碼的相容性。

伍、(一)、本案代表圖為：第 4 圖

六、英文發明摘要（發明名稱：Firmware Updating Method And Related Apparatus For Checking Content of Replacing Firmware Before Firmware Updating）

A method and related apparatus for firmware updating of an electronic system. The electronic system includes a host and a peripheral device, the device has a control circuit and a flash memory for storing a first firmware code. The control circuit executes the first firmware code to control the device according to control commands from the host. The method includes:



四、中文發明摘要（發明名稱：以檢查韌體程式內容確保韌體更新相容性之韌體更新方法及相關裝置）

(二)、本案代表圖之元件代表符號簡單說明：

200 流程 202-224 步驟

六、英文發明摘要（發明名稱：Firmware Updating Method And Related Apparatus For Checking Content of Replacing Firmware Before Firmware Updating）

before replacing/updating the first firmware code with a second firmware code, performing a checking step to check whether the content of the second firmware code matches a predetermined content to ensure compatibility of the second firmware code. The checking step could be performed by the host and/or by the control circuit. The checking step could be performed by



四、中文發明摘要 (發明名稱：以檢查韌體程式內容確保韌體更新相容性之韌體更新方法及相關裝置)

六、英文發明摘要 (發明名稱：Firmware Updating Method And Related Apparatus For Checking Content of Replacing Firmware Before Firmware Updating)

checking whether values/strings of constants defined in the second firmware code match predetermined values/strings, and/or by checking whether commands/information in predetermined addresses of the second firmware code match predetermined commands/information.



一、本案已向

國家(地區)申請專利

申請日期

案號

主張專利法第二十四條第一項優先權

無

、主張專利法第二十五條之一第一項優先權：

申請案號：

無

日期：

三、主張本案係符合專利法第二十條第一項第一款但書或第二款但書規定之期間

日期：

四、有關微生物已寄存於國外：

寄存國家：

無

寄存機構：

寄存日期：

寄存號碼：

有關微生物已寄存於國內(本局所指定之寄存機構)：

寄存機構：

無

寄存日期：

寄存號碼：

熟習該項技術者易於獲得，不須寄存。



五、發明說明（1）

發明所屬之技術領域：

本發明係提供一種韌體更新的方法及相關裝置，尤指一種在韌體更新前檢查更新用韌體程式碼之內容以判斷其相容性、適用性的方法及相關裝置。

先前技術：



五、發明說明 (2)

實際運作。舉例來說，電腦系統中的光碟機本身就具有其控制電路及對應的快閃記憶體，以儲存光碟機本身的韌體程式碼。當電腦的主機要讀取一光碟片上的資料時，僅需向光碟機指定該筆資料在光碟片上的位址，光碟機的控制電路就會執行本身的韌體程式碼，協調光碟機中諸如馬達 (spindle)、雷射讀取頭 (pick-up head) 等等機電元件之運作（像是馬達要達到特定的轉速，讀取頭要移動、鎖定至特定的位置，接收光碟片上反射的雷射等等），實際達成主機的要求。

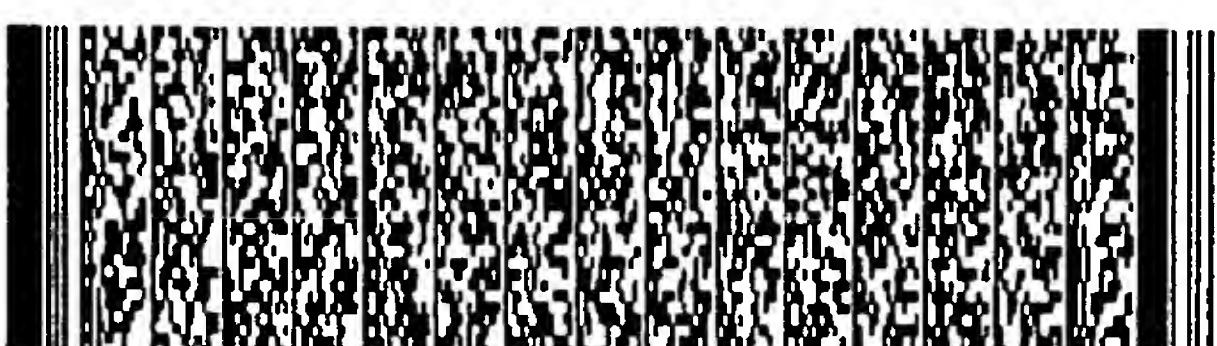
請參考圖一。圖一為一典型電子系統 10 的功能方塊示意圖。電子系統 10 可以是一電腦系統，其中包括有一主機 12 及一或多個周邊裝置（圖一中繪出兩個周邊裝置 14、15 做為代表）。周邊裝置可以是唯讀光碟機、光碟燒錄機、硬碟、外接式的快閃記憶體等等；以周邊裝置 14 為例來說明典型周邊裝置中的配置，周邊裝置 14 中設有一控制電路 16、一緩衝記憶體 18、一儲存記憶體 20 及一伺服硬體 22。若電子系統 10 為一電腦系統，則主機 12 的典型配置就如圖一中所示，其設有一中央處理器 26、一北橋電路 28A、一南橋電路 28B 及一揮發性的記憶體 30、繪圖卡 32A 及顯示器 32B。中央處理器 26 用來主控主機 12 的運作，記憶體 30 用來暫存中央處理器 26 運作期間必需的數據、資料，繪圖卡 32A 則用來處理影像資料，以將主機 12 運作的狀況顯示於顯示器 32B 上。而北橋電路



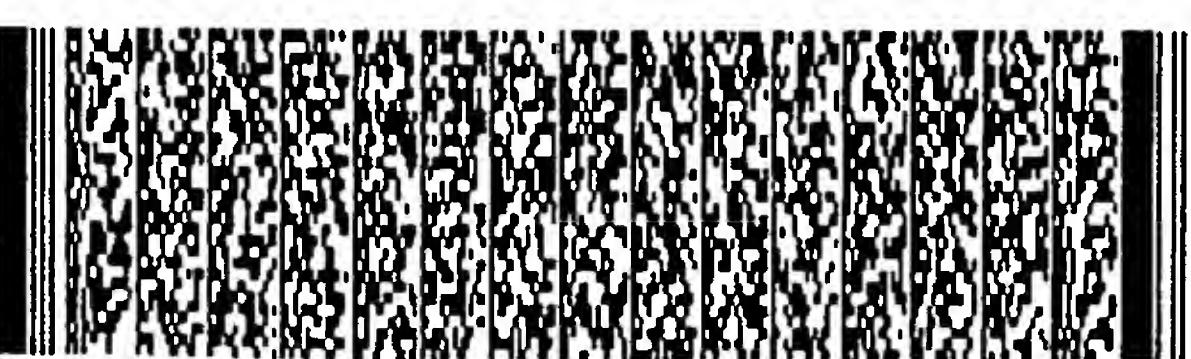
五、發明說明 (3)

28A即用來管理、控制中央處理器26與繪圖卡32A、記憶體30的資料傳輸；南橋電路28B則透過北橋電路28A電連於中央處理器26，作為各周邊裝置14、15與主機12交換資料的介面。透過南橋電路28B與周邊裝置14、15間的電連（像是透過匯流排），主機12與各周邊裝置14、15就能互相交換資料。另一方面，在周邊裝置14中，控制電路16用來接收主機12傳來的控制指令，以控制周邊裝置14的運作。伺服硬體22即是受控於控制電路16，用來實現周邊裝置14的功能。舉例來說，若周邊裝置14為一光碟機，伺服硬體22即包括了帶動光碟片轉動的馬達、雷射讀取頭等機電元件，以存取一光碟片上的資料。緩衝記憶體18、儲存記憶體20則用來支援控制電路16的運作；其中緩衝記憶體18為一揮發性的記憶體（像是隨機存取記憶體），用來暫存控制電路16運作期間的數據、資料。儲存記憶體20則為一非揮發性的記憶體（像是快閃記憶體），用來記錄一韌體的程式碼24；正如前面所提到的，周邊裝置14的控制電路16在操控周邊裝置14時，會依循特定的韌體程式碼來進行各種操控程序，程式碼24即為此韌體程式碼，用來記錄不同操控程序進行的方式。控制電路16執行此程式碼24，就能根據主機12的操控指令，控制周邊裝置14的運作。

當然，在每一個周邊裝置出廠時，儲存記憶體中均已寫入韌體程式碼，作為預置的韌體。在周邊裝置連接



五、發明說明 (4)



一般來說，要更新周邊裝置中的韌體程式碼，主機也要進行對應的運作。請參考圖二（並一併參考圖一）。圖二流程圖中之流程 100，即為習知技術中，電子系統 10 要對周邊裝置 14 進行韌體更新時，主機 12、周邊裝置 14 間協調運作之流程。圖二左半邊的步驟代表主機 12 進行的步驟，右半邊的步驟代表周邊裝置 14 所進行的步驟。習知流程 100 中依序有下列步驟：

步驟 102：開始。要開始更新周邊裝置 14 的韌體程式碼時，中央處理器 26 會開始執行一韌體更新的應用程式 34，啟動整個韌體更新之流程 100，並在後續步驟中持續控制韌體更新流程之進行。就如圖一中所示，此應用程式 34 會被載入至主機 12 中的記憶體 30，由中央處理器 26 執行應用程式 34，對周邊裝置 14 進行韌體更新。韌體更新之目的，是以一新程式碼 36，取代周邊裝置 14 中既有的程式碼 24。

步驟 104：主機 12 執行應用程式 34 後，就會先對周邊裝置 14 進行裝置識別，以確定應用程式 34 是和周邊裝置 14 搭配的應用程式，能正確地控制周邊裝置 14 進行韌體更新。如前所述，主機 12 可能連接有多個不同的周邊裝置，各周邊裝置的結構、功能各不相同，要操控不同的周邊裝置分別進行韌體更新，主機 12 所要執行的應用程式當然也各不相同；要正確地對一周邊裝置進行韌體更新，就要使用和該周邊裝置對應的應用程式。而本步驟

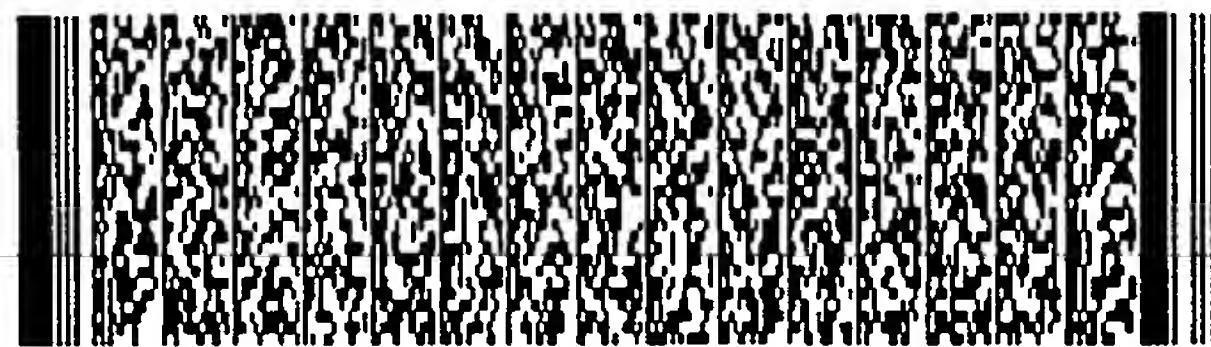
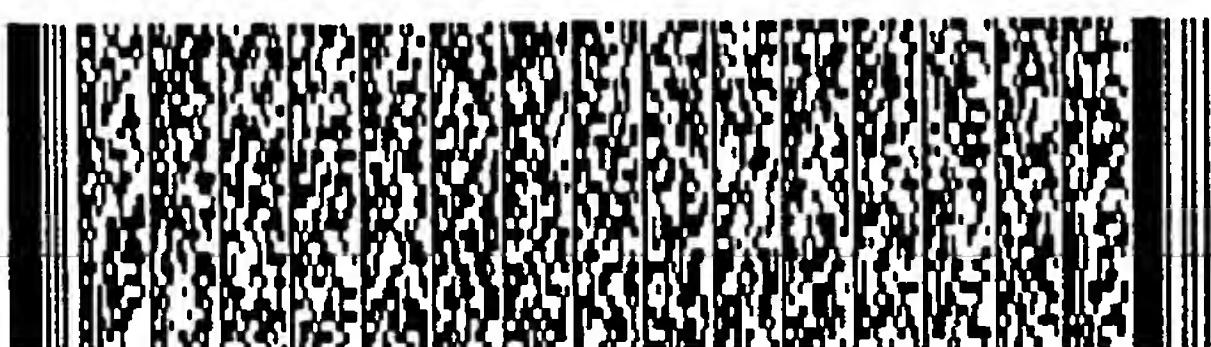


五、發明說明 (6)

就是要確定應用程式 34是和周邊裝置 14對應的正確應用程式。一般來說，各周邊裝置中的韌體程式碼內就記錄有一韌體識別碼，其中可包括有韌體研發廠商的名稱 (vendor ID)、該韌體程式碼適用之周邊裝置的型號名稱 (model name)，或是該韌體程式碼的版本資訊等等。像是在周邊裝置 14中的既有程式碼 24，就記錄有一韌體識別碼 24I。而當主機 12在執行應用程式 34時，就可依據應用程式 34的指示向周邊裝置 14發出裝置識別的控制指令，要求周邊裝置 14將韌體識別碼 24I中之相關資料（或其他可識別周邊裝置 14的資料、訊號）傳輸至主機 12。

步驟 106：周邊裝置 14回應主機 12於步驟 104中的要求，將韌體識別碼 24I之相關資料（或其他識別資料）回傳至主機 12。

步驟 108：主機 12執行應用程式 34時，即可根據周邊裝置 14回傳的識別資料，判斷周邊裝置 14是否是和應用程式 34搭配的裝置。事實上，在主機 12與周邊裝置 14進行裝置識別時，很有可能會進行數次的資料、指令交換，在不影響本發明技術討論的情形下，圖二中的流程圖已經部分簡化裝置識別的細節。根據周邊裝置 14與主機 12間的資料交換，若主機 12判斷出應用程式 34的確能和周邊裝置 14互相搭配，主機 12即可繼續韌體更新的流程，發控制指令訊問周邊裝置 14現行的運作狀態是否能進行韌體更新。因為在主機 12執行韌體更新之應用程式 34時，周邊裝置 14可能正在執行某些運作（譬如說光碟機



五、發明說明 (7)

之周邊裝置 14可能正在存取光碟片上的資料），這樣就不能進行韌體更新了。故此步驟中，主機 12要詢問周邊裝置 14目前的運作狀態。同時，主機 12也可將韌體更新所用的新程式碼 36載入至記憶體 36中，準備傳輸給周邊裝置 14。

步驟 110：周邊裝置 14回應主機的訊問，將周邊裝置 14目前的運作狀態回傳至主機 12。

步驟 112：若周邊裝置 14回傳之運作狀態代表其能進行韌體更新，主機 12就能將暫存於記憶體 30中的新程式碼 36傳輸至周邊裝置 14。就像典型的網路傳輸一樣，主機 12在傳輸新程式碼 36時，會執行一預設的檢查碼產生演算法，根據新程式碼 36的內容計算出一檢查碼 (checksum)36C，並與新程式碼 36一併傳輸至周邊裝置 14。

步驟 114：周邊裝置 14接收主機 12傳來的新程式碼及檢查碼，並將其暫存於緩衝記憶體 18中。圖一中暫存於周邊裝置 14緩衝記憶體 18內的新程式碼 37、檢查碼 37C，就是周邊裝置 14由主機 12接收的新程式碼及檢查碼。

步驟 116：控制電路 16同樣也會執行檢查碼產生演算法，根據其所接收到的新程式碼 37計算出另一檢查碼 39C，並將此檢查碼 39C和由主機 12傳來的檢查碼 37C進行比對。控制電路 16使用的檢查碼產生演算法和主機 12根據新程式碼 36產生檢查碼 36C的檢查碼產生演算法是一樣的；若在新程式碼、檢查碼由主機 12傳輸至周邊裝置 14的過程

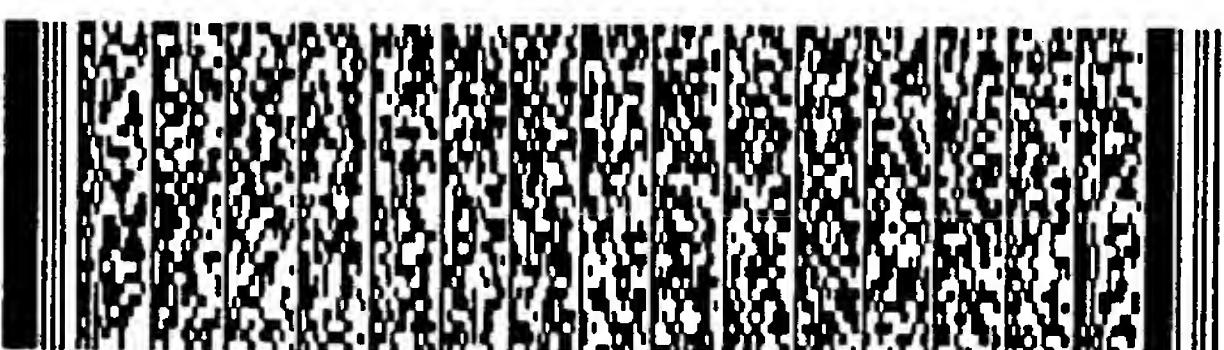


五、發明說明 (8)

中沒有發生資料傳輸的錯誤，周邊裝置 14接收到的新程式碼 37應該就等於主機 12所要傳輸的新程式碼 36，而控制電路 16所計算出來的檢查碼 39C應該也會等於其所接收的檢查碼 37C。反之，若控制電路 16計算出來的檢查碼 39C其接收到的檢查碼 37C不一致，就代表主機 12傳輸新程式碼 / 檢查碼至周邊裝置 14的過程中發生了資料傳輸的錯誤。若在本步驟中比對出檢查碼 39C與檢查碼 37C為一致的，就可進行至步驟 118；反之，若兩檢查碼不一致，就要進行至步驟 120。

步驟 118：在習知技術中，控制電路 16在步驟 116中判斷新程式碼 37的傳輸無誤後，就會將儲存記憶體 20中既有程式碼 24抹除 (erase)，並將暫存於緩衝記憶體 18中的新程式碼 37覆寫至儲存記憶體 20中；這樣一來也就完成周邊裝置 14的韌體更新了。接下來控制電路 16就可以執行被寫入至儲存記憶體 20的新程式碼 37，以控制周邊裝置 14的運作。當然，在韌體更新成功後，周邊裝置 14也可將更新成功之情況回傳至主機 12；而主機 12也可再次要求周邊裝置 14傳輸新程式碼中的韌體識別碼至主機 12以確認韌體更新成功。在不影響本發明技術討論的情形下，圖二中已略去上述運作之細節。

步驟 120：若在步驟 116中控制電路 16比對發現檢查碼 37C、39C不一致，就可進行必要的錯誤處理。舉例來說，控制電路 16可要求主機 12將新程式碼 36重新傳輸一次，重新進行步驟 116中的檢查碼比對；或是將發生錯誤

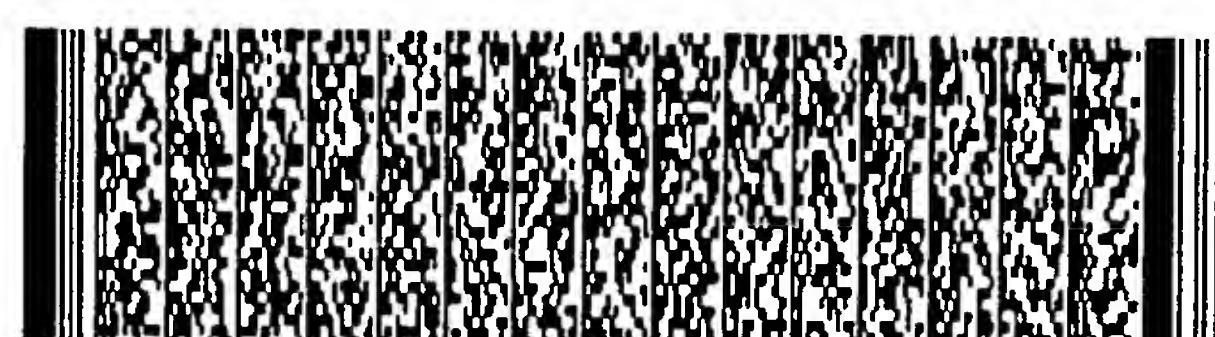


五、發明說明 (9)

的情況回傳至主機 12，由主機 12 決定後續的運作。

步驟 122：結束韌體更新的流程。

總結來說，在以圖二中之習知流程 100 進行韌體更新時，主機 12、周邊裝置 14 間在確認互連之狀態、周邊裝置 14 可以進行韌體更新後，主機 12 只是單純將韌體更新之新程式碼 36 傳輸至周邊裝置 14，而周邊裝置 14 在進行步驟 116 的檢查碼比對而確認新程式碼之傳輸無誤後，就會進行韌體更新。上述習知流程的主要缺點之一，就是無法確保韌體更新用之新程式碼的內容是否真的是合乎周邊裝置 14 應用的韌體程式碼。一般來說，韌體更新之程式碼 36 是由主機 12 的使用者取得（像是由網路下載），再於韌體更新的過程中由應用程式 34 載入至主機 12 的記憶體 30 中。然而，在使用者取得新程式碼 36 的過程中就有可能出錯，使得新程式碼 36 本身為不符合周邊裝置 14 應用的韌體程式碼。舉例來說，主機 12 的使用者可能是由網路下載而取得新程式碼 36，但網路下載的過程中可能會發生網路傳輸的錯誤，使得新程式碼 36 本身是不完整的。或者，使用者也可能誤判韌體程式碼的版本。如前所述，韌體研發廠商可能不斷釋出新版本的韌體程式碼，周邊裝置 14 在出廠時可能已經內建了較新版本的預置程式碼，但使用者在未察覺的情況下，卻取得了較舊版本的新程式碼 36 要來為周邊裝置 14 進行韌體更新；換句話說，在此情況下，圖一中的既有程式碼 24 為



五、發明說明 (10)

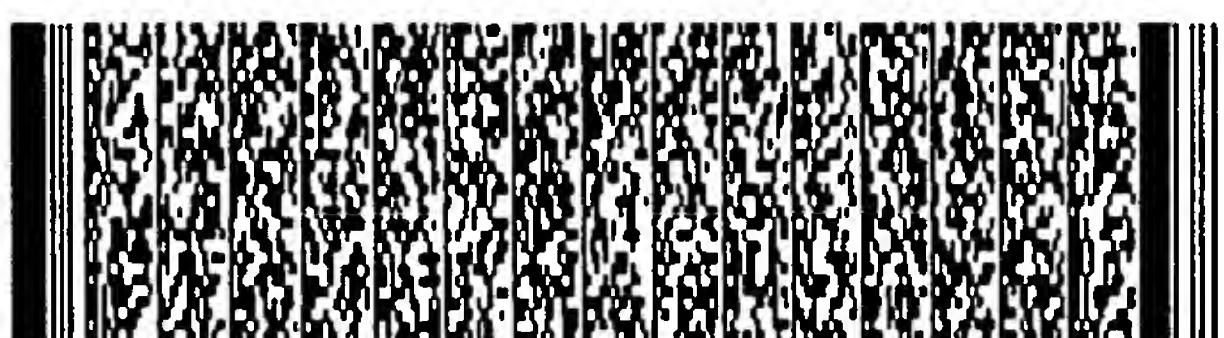
較新的版本，新程式碼 36 本身反而是較舊的版本。此時若進行韌體「更新」，反而會將韌體程式碼降級。另外，使用者也可能錯誤地取得不合乎周邊裝置 14 應用的韌體程式碼作為新程式碼 36。舉例來說，使用者取得之程式碼 36 可能根本不是適用於周邊裝置 14 的韌體程式碼。尤其是在技術發展快速的現代社會，同一廠商可能是推出多種不同型號、功能各異的同類型周邊裝置。像這種在光碟機這種常見的電腦系統周邊裝置中，就有各種不同讀取倍速的光碟機，或是某些光碟機僅能取光碟片資料，有些光碟機則還能將資料燒錄至光碟片；某些光碟機則支援較多不同格式光碟片之資料讀取。雖然研發廠商會針對不同型號的周邊裝置釋出不同的更新韌體程式碼，但使用者可能一時不察，取得與周邊裝置 14 完全不相符的新程式碼 36 來為周邊裝置 14 進行韌體更新。除此之外，還有可能有某些惡意的破壞者，故意提供錯誤的韌體程式碼作為新程式碼 36，意圖以韌體植入周邊裝置 14 中，以癱瘓周邊裝置 14 的正常運作。

在上述的各種情況下，主機 12 要用來進行韌體更新的新程式碼 36，其實已經不合乎周邊裝置 14 的應用；但習知的韌體更新流程 100 中，卻無法檢查出新程式碼 36 是否適用。在習知流程 100 中，主機 12 雖然會在步驟 104 進行裝置識別，透過周邊裝置 14 檢查既有之程式碼 24 的



五、發明說明 (11)

韌體識別碼 24I，但主機 12 檢查的是周邊裝置 14 已在運用中的韌體程式碼（也就是周邊裝置 14 既有之程式碼 24），不會檢查新程式碼 36 是否合乎周邊裝置 14 的應用。另外，周邊裝置 14 的控制電路 16 會在步驟 116 中檢查與新程式碼 36 相關的檢查碼，但此步驟只能檢查出新程式碼 36 在主機 12、周邊裝置 14 間傳輸時是否發生資料傳輸的錯誤，也不能檢查出新程式碼 36 是否適用。即使新程式碼為不適用的韌體程式碼，只要主機 12、周邊裝置 14 間的資料傳輸沒有發生問題，流程 100 還是會通過步驟 116，將不適用的新程式碼寫入至儲存記憶體中。換句話說，在步驟 116 中，即使檢查碼符合，也只代表主機 12 是不適用的新程式碼 36 完整正確地傳輸至周邊裝置 14 而成為新程式碼 37（見圖二），新程式碼 37 不適用之本質還是不會改變，也不能由步驟 116 中檢查出來。雖然控制電路 16 在步驟 116 中會依據新程式碼 37 計算出檢查碼 39C，使得檢查碼 39C 能反映出新程式碼 37 的內容，但步驟 116 中用來比對檢查碼 39C 的另一檢查碼 37C 也還是依據新程式碼 36 所算出的，並不能代表「適用」之韌體程式碼所應具有的檢查碼（尤其是當新程式碼 36 本身就是不適用的韌體程式碼時）。換句話說，當習知流程 100 在進行步驟 116 時，控制電路 16 本身並不能得知「適用」之韌體程式碼對應之檢查碼為何，當然也就不能在步驟 116 中檢查比對出新程式碼 37 是否適用。若在韌體更新流程中誤將錯誤、不適用的韌體程式碼植入至周邊裝置 14 中，



五、發明說明 (12)

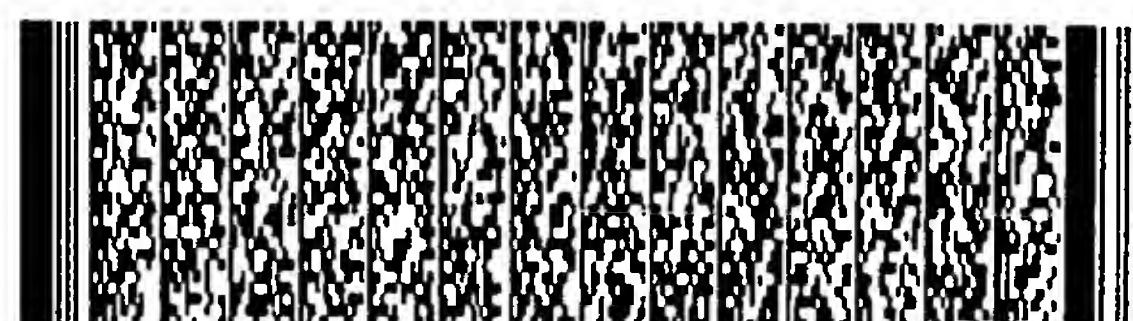
不但無法達到韌體更新的目的，還會導致周邊裝置 14 不能正確運作的嚴重後果。

發明內容：

因此，本發明之主要目的，在於提供一種新的韌體更新方法及相關裝置，能篩選出更新用之新韌體程式碼是否真正適用，以克服習知技術的缺點。

在習知技術中，不論是在主機端或是在周邊裝置的裝置端，都無法檢查出韌體更新所用之新程式碼是否合乎周邊裝置的實際應用，因此也無法避免不適用之新程式碼在習知之韌體更新流程中被誤植入周邊裝置。

在本發明中，則會在對一周邊裝置進行韌體更新的流程中加入主機端檢查步驟及 / 或裝置端檢查步驟，將更新用之新韌體程式碼的部分內容與一預設之內容比對，以實際判斷新韌體程式碼是否適用。在實際實施時，可商檢的新韌體程式碼中的韌體程式碼（包括裝置型號識別碼、廠名稱、商標名稱，以及該周邊裝置中既有之韌體程式碼）是否和該周邊裝置中既存之韌體程式碼中是否含有特定定位址的指令、常數。另外，也可檢查新韌體程式碼中特定定位址的指令、資料是否合乎預期中的指令、資料，或是搜尋新



五、發明說明 (13)

韌體程式碼中特定指令、資料的位址是否位於預設之位址，以判斷韌體更新所用的新韌體程式碼是否是合乎周邊裝置應用的正確韌體。上述的檢查步驟可在主機端、裝置端分別獨立進行，確保韌體更新所用的新韌體程式碼為適用的韌體程式碼，避免在韌體更新流程中誤將不適用的韌體程式碼植入周邊裝置中。

實施方式：

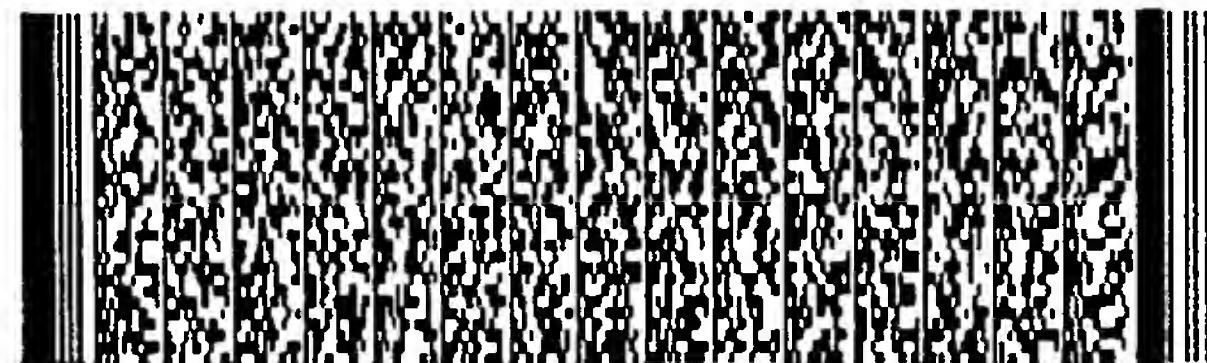
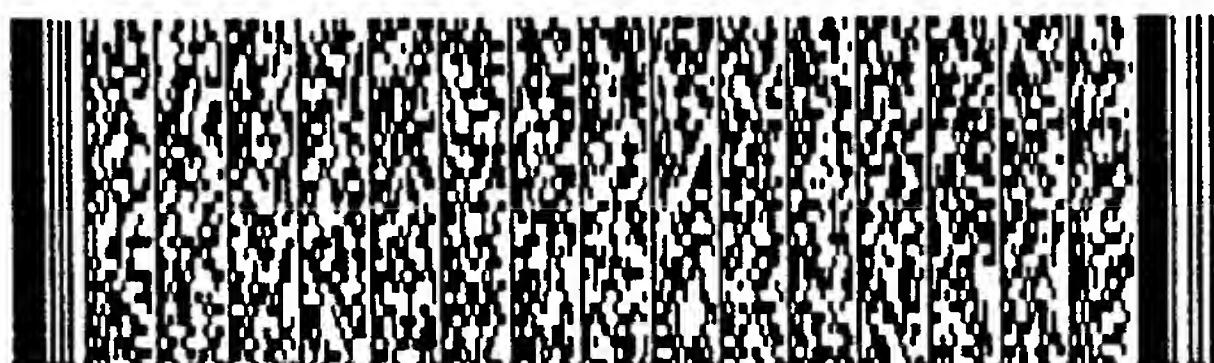
請參考圖三。圖三為本發明電子系統 50 配置之功能方塊示意圖。電子系統 50 中設有一主機 52 及一或多個搭配運作的周邊裝置（圖三中繪出兩個周邊裝置 54、55 做為代表），以擴充主機 52 本身的功能。電子系統 50 可以是一電腦系統；在此情況下，主機 52 中可設有中央處理器 66、北橋電路 68A、南橋電路 68B、記憶體 70、繪圖卡 72A 及顯示器 72B；周邊裝置 54、55 則可以是光碟機、光碟燒錄機、硬碟機等等。以周邊裝置 54 為例來說明周邊裝置中的配置；周邊裝置 54 中設有一控制電路 56、一用來實現周邊裝置 54 功能的伺服硬體 62、一用來以揮發性方式暫存資料的緩衝記憶體 58（像是隨機存取記憶體），以及一用來以非揮發性的方式儲存資料的儲存記憶體 60（像是快閃記憶體）；而控制電路 56 中則有一檢查模組 56B。在主機 12 中，中央處理器 66 用來主控主機 52 的運作，繪圖卡 72A 能將主機 52 運作之狀態與結果轉變為影



五、發明說明 (14)

像訊號，並於顯示器 72B顯示出來。揮發性的記憶體 70（像是隨機存取記憶體）用來暫存中央處理器 66運作時必需的程式碼及相關資料、數據。北橋電路 68A即用來管理中央處理器 66、記憶體 70及繪圖卡 72A間的資料傳輸。透過電連於北橋電路 68A的南橋電路 68B，主機 12就能和各周邊裝置 54、55交換指令、資料；其中南橋電路 68B和各周邊裝置間可以用匯流排（像是 IDE、EIDE等匯流排）相互連接。在周邊裝置 54方面，就如前面討論過的，為了要操控周邊裝置 54中進行多種運作，也會以一韌體程式碼來記錄各種操控程序實行的方法；圖三中儲存於儲存記憶體 60中的既有程式碼 64就是此一韌體程式碼。控制電路 56接收主機 52傳來的控制指令後，就能執行儲存記憶體 60中的程式碼 64來操控伺服硬體 62，以實現主機 52要求的功能。而緩衝記憶體 58則用來暫存裝置 54周邊裝置 54為一光碟燒錄機，伺服硬體 62會暫存於緩衝記憶體 58中，再由伺服硬體 62實際將這些暫存於緩衝記憶體 58的資料燒錄至光碟片上；而伺服硬體 62由光碟片上電路 56安排，將這些資料回傳至主機 52。

就如前面所討論過的，周邊裝置會有韌體更新的需要。請參考圖四（並一併參考圖三）。圖四所示即為本



五、發明說明 (15)

發明於電子系統 50中進行韌體更新之流程 200；於圖四圖面左方的步驟是在主機 52這一端進行，右方的步驟是由周邊裝置 54進行。流程 200中設有下列步驟：

步驟 202：開始。當電子系統 50要對周邊裝置 54進行韌體更新時，主機 52中的中央處理器 66就會將一韌體更新之應用程式 74載入至記憶體 70中（請一併參考圖三），並開始執行應用程式 74，以啟動整個韌體更新之流程 200，並在後續步驟中持續控制流程 200的進行。韌體更新之目的，是要以一新程式碼 76，取代周邊裝置 54中既有之韌體程式碼 64。

步驟 204：主機 52要求周邊裝置 54進行裝置識別。如前所述，各周邊裝置的韌體程式碼中，都記錄有一韌體識別碼，記載韌體研發廠商的名稱 (vendor ID)，以及該韌體程式碼適用之周邊裝置的型號名稱 (model name)。就如圖三中所示，周邊裝置 54在韌體更新前的既有程式碼 64，也記錄有對應程式碼 64的韌體識別碼 64I。在進行裝置識別時，主機 52即可發出控制指令，要求周邊裝置 54回傳韌體識別碼 64I的相關資料，以確認應用程式 74是和周邊裝置 54搭配的正確程式，能在後續步驟中正確地和周邊裝置 54配合以進行韌體更新。

步驟 206：周邊裝置 54中的控制電路 56接收到主機 52於步驟 204中傳來的控制指令後，就會將韌體程式碼 64中與韌體識別碼 64I相關之資料回傳至主機 52。

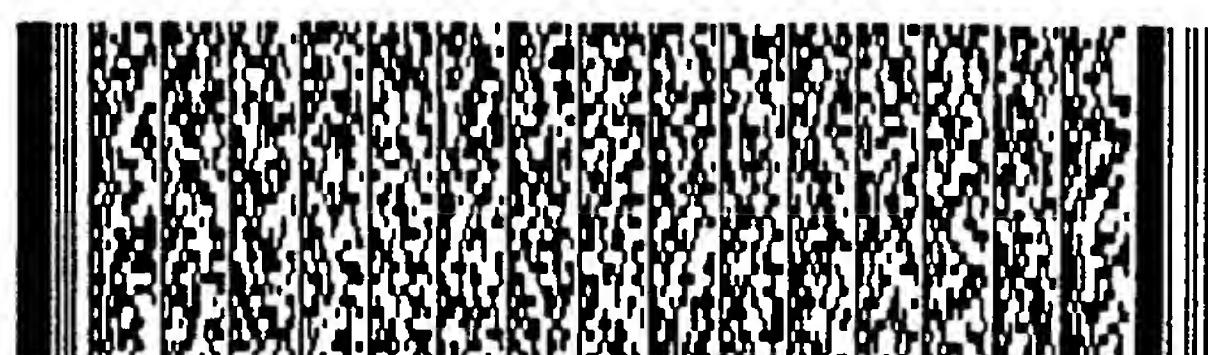
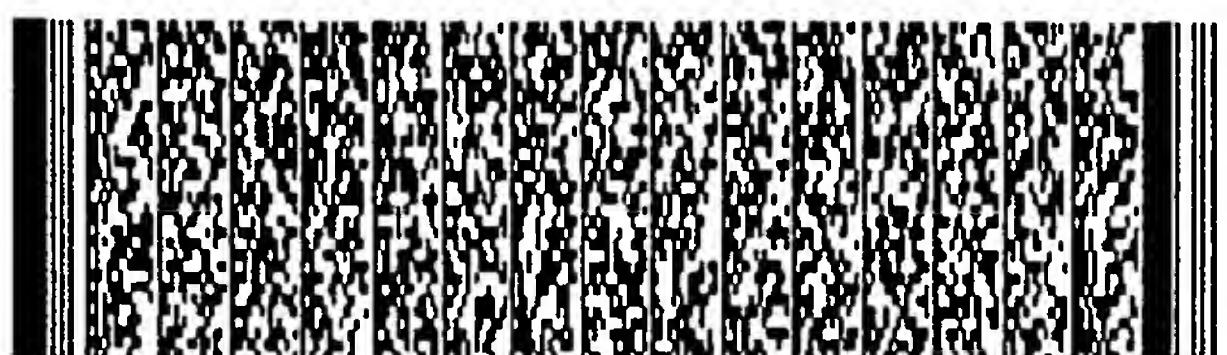
步驟 208：主機 52根據周邊裝置 54回傳的韌體識別碼 64I



五、發明說明 (16)

資料（或其他主機 52 要求的識別資料），就可以判斷應用程式 74 能否配合周邊裝置 54 的韌體更新。若主機 52 確認應用程式 74 的確能和周邊裝置 54 搭配，主機 52 就會繼續執行應用程式 74，以進行周邊裝置 54 韌體更新的後續新實步驟。在此同時，中央處理器 66 也可將周邊裝置 54 韌體更新之在進行程式碼 76 載入至記憶體 70 中（就如圖三中所示）。在進行施裝置識別的過程中，主機 52、周邊裝置 54 可能會有數次的資料交換；在不影響本發明技術揭露的情形下，圖四中已適當地省略了裝置識別的細節。

除了進行裝置識別之外，為了進一步確認新程式碼之內容為適合周邊裝置 54 應用的韌體程式碼，本發明還會額外進行一主機端檢查步驟，以依據新程式碼 76 的內容，判斷新程式碼 76 是否為適用的韌體程式碼。主機端檢查步驟進行的方式可有數種實施例。舉例來說，新程式碼 76 既然為一韌體程式碼，其中當然也記載了一韌體識別碼 76I（請一併參考圖三），就像周邊裝置 54 中的既有程式碼 64 也有一對應的韌體識別碼 64I。而在此主機端檢查步驟中，主機 12 就能檢查新程式碼 76 中的韌體識別碼 76I 是否和既有程式碼 64 中的韌體識別碼 64I 相符，以驗證新程式碼 76 是否和周邊裝置 54 中既有之程式碼 64 一樣由同一韌體研發廠商所研發，或是適用於相同型號的周邊裝置。另外，韌體研發廠商也能事先在韌體程式碼中的固定位址記錄預設的控制指令或字串、資料，形

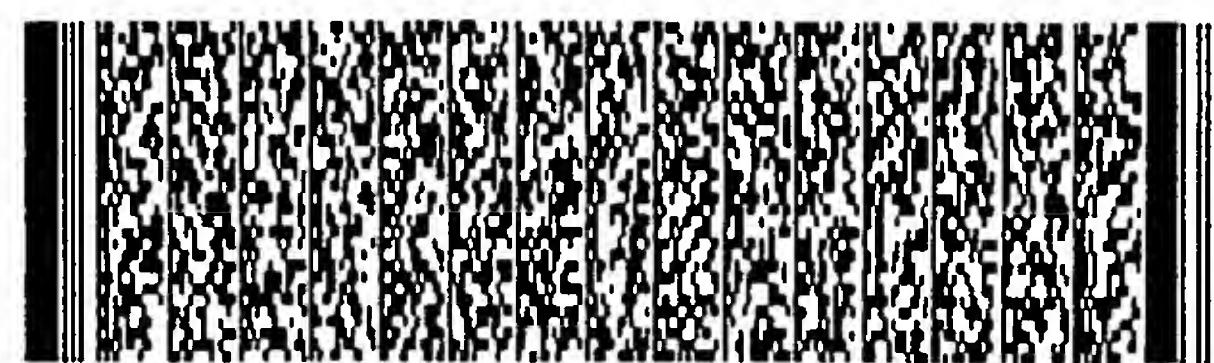


五、發明說明 (17)

成一預設內容 80(見圖三)；等主機 52進行主機端檢查步驟時，就能檢查新程式碼 76在該固定位址中是否具有預設內容 80(如圖三所示意的)，以判斷新程式碼 76的適用性。或者，主機 52也可搜尋新程式碼 76中具有預設內容的部分，再檢查該部分的位址是否在韌體研發廠商預設之位址，以判斷新程式碼 76的適用性。主機端檢查步驟進行的各種方式及原理，會在稍後更進一步討論。若進行主機端檢查步驟後主機 52判斷新程式碼 76為一適用的正確程式碼，就能繼續韌體更新的流程。接下來主機 52可發出指令詢問周邊裝置 54的狀態是否可進行韌體更新。

步驟 210：周邊裝置 54回應主機於步驟 208中的訊問，將周邊裝置 54目前的狀態回傳至主機 52。

步驟 212：主機 52接收周邊裝置 54於步驟 210中回應的訊號；若周邊裝置 54正處於可進行韌體更新的狀態，主機 52就可以開始將已載入至記憶體 70的新程式碼 74傳輸至周邊裝置 54。如同步驟 204、206中的裝置識別，步驟 210、212中對周邊裝置 54的裝置狀態檢查也可能會在主機 52、周邊裝置 54間進行數次的資料往來傳輸；在不影響本發明技術揭露之情況下，圖四中已適當地簡化相關細節。就如圖三中所示意的，在主機 52要將新程式碼 76傳輸至周邊裝置 54之前，也會以一預設之檢查碼產生演算法，根據新程式碼 76的內容計算出一檢查碼 76C，並將其附加於檢查碼 76，連同檢查碼 76一起傳輸至周邊裝置



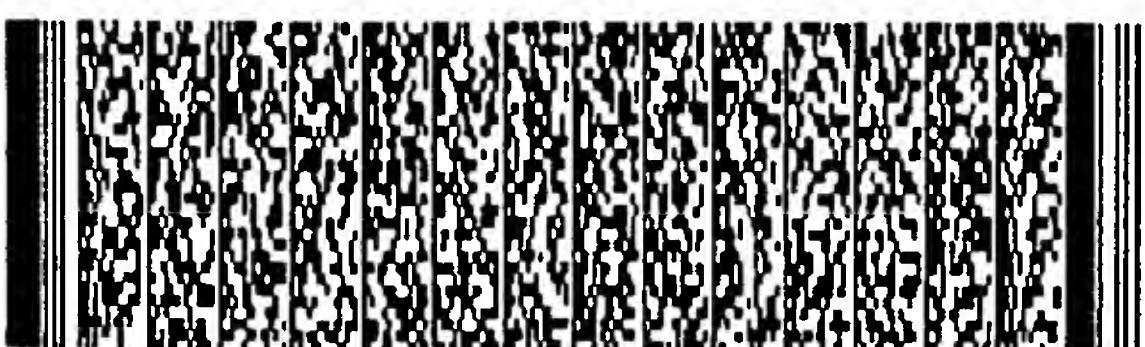
五、發明說明 (18)

54。

步驟 214：周邊裝置 54接收主機 52傳來的新程式碼及附加的檢查碼，並將其暫存於緩衝記憶體 58中，也就是圖三中所示的新程式碼 77及檢查碼 77C。

步驟 216：周邊裝置 54的控制電路 56會利用檢查碼產生演算法，根據新程式碼 77的內容計算出一檢查碼 79C(見圖三)，並驗證檢查碼 79C與周邊裝置 54由主機 52處接收到的檢查碼 77C是否相同。若兩者相同，代表主機 52在將新程式碼傳輸至周邊裝置 54的過程中沒有發生資料傳輸錯誤。

在確定韌體更新用之新程式碼已經完整地由主機 52傳輸至周邊裝置 54後，本發明還會在周邊裝置 54中另外進行一裝置端檢查步驟，由控制電路 56實現檢查模組 56B的功能，以檢查暫存於緩衝記憶體 58中的新程式碼 77是否為適用的韌體程式碼。舉例來說，控制電路 56可比較暫存於緩衝記憶體 58中的新程式碼 77是否和儲存記憶體 60中的既有程式碼 64具有相同的韌體識別碼。由於新程式碼 77是由主機 52傳輸過去的，在資料傳輸無誤的情形下，新程式碼 77也具有和韌體識別碼 76I相同的韌體識別碼 77I；控制電路 56比對韌體識別碼 77I和既有韌體程式 64的韌體程式碼 64I是否相符，就能判斷新程式碼 77是否為適用於周邊裝置 54的韌體程式碼。類似於主機端檢查步驟，於周邊裝置 58中進行的裝置端檢查步驟也可以



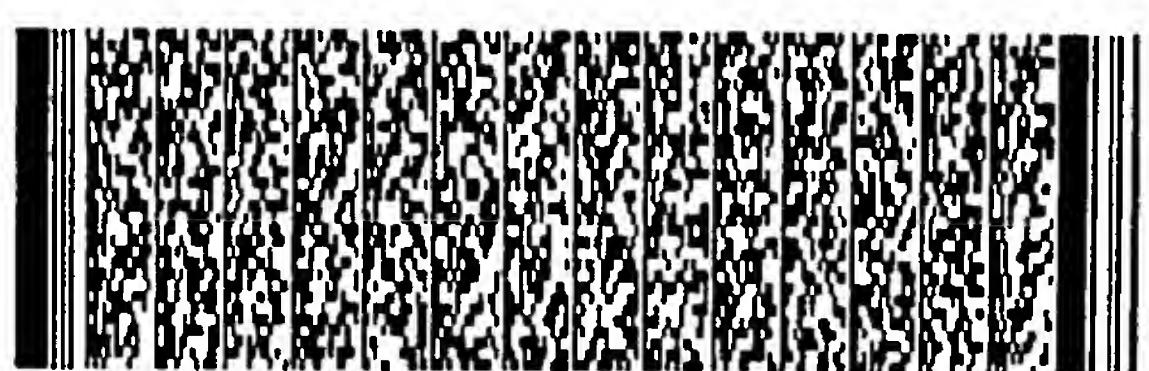
五、發明說明 (19)

檢查新程式碼 77 中位於特定之預設位址的部分內容是否符合一預定內容 82 (如圖三中所示意的)；或是在新程式碼 77 中搜尋某些預設內容是否存在，或是否位於某些特定位址。至於本發明裝置端檢查步驟進行的細節、原理，將在後面做進一步的說明。

當然，若在進行本發明裝置端檢查步驟前，控制電路 56 已發現檢查碼 77C 與控制電路 56 計算出來的檢查碼 79C 不符合，代表新程式碼再由主機 52 傳輸至周邊裝置 54 的過程中發生資料傳輸的錯誤。此時控制電路 56 可將錯誤的情形回傳至主機 52，或要求主機 52 重新傳輸新程式碼，直到新程式碼被完整的傳輸至周邊裝置 54，再進行本發明之裝置端檢查步驟。

步驟 218：若控制電路 56 進行裝置端檢查步驟後判斷新程式碼 77 的確適用於周邊裝置 54，就能進行至步驟 220；若否，則進行至步驟 222。

步驟 220：通過了檢查碼的驗證，以及本發明裝置端檢查步驟的驗證，周邊裝置 54 一方面能確認本身已正確接收由主機 52 傳來的新程式碼 77，另一方面也能確認新程式碼 77 為周邊裝置 54 適用的韌體程式碼，此時控制電路 56 就能將儲存記憶體 60 中原先做為韌體程式碼的程式碼 64 擦除，將新程式碼 77 寫入至儲存記憶體 60 中，以取代原先的程式碼 64，完成韌體更新。接下來控制電路 60 就能執行儲存記憶體 60 中的新程式碼 77，以新的操控程序操



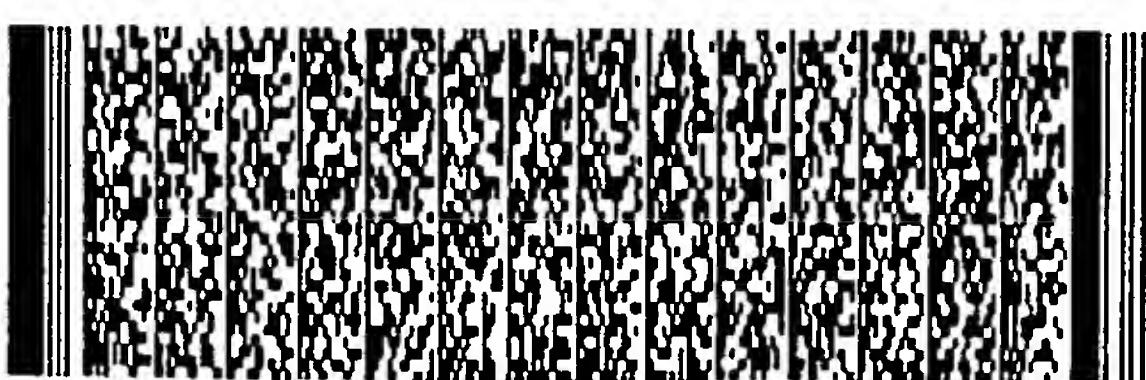
五、發明說明 (20)

控周邊裝置 54 的運作。當然，在韌體更新完成後，周邊裝置 54 也能向主機 52 回報韌體更新完成之狀況，主機 52 也可要求周邊裝置 54 再度將新韌體程式碼中韌體識別碼之相關資訊回傳，確認韌體更新已完成等等。在不影響本發明技術揭露的情形下，這部分的細節流程已於圖四中省略。

步驟 222：若在步驟 216 中進行本發明之裝置端檢查時，發現新程式碼 77 並非適用的韌體程式碼，控制電路 56 就要進行錯誤處理。控制電路 56 可將新程式碼不適用的情形回傳至主機 52，由主機 52 的使用者決定後續的步驟。最重要的是，此時控制電路 56 不會將不適用的新程式碼覆寫至儲存記憶體 60，使得控制電路 56 在後續的過程中不會依據不適用的新程式碼來操控周邊裝置 54，而周邊裝置 54 的運作也不會因韌體更新流程中的不適用韌體程式碼而受影響。

步驟 224：結束本發明韌體更新之流程。

由以上對本發明韌體更新流程之描述可知，本發明之流程 200 除了在韌體更新的過程中進行了裝置識別、檢查碼之確認外，還額外地進行了主機端檢查步驟及裝置端檢查步驟。前者能在主機 52 在將韌體更新之新程式碼 76 傳輸至周邊裝置 54 之前，先檢查新程式碼 76 是否為適用於周邊裝置 54 的韌體程式碼。在將新程式碼 76 順利傳輸至周邊裝置 54 而成為新程式碼 77 後，周邊裝置 54 還會



五、發明說明 (21)

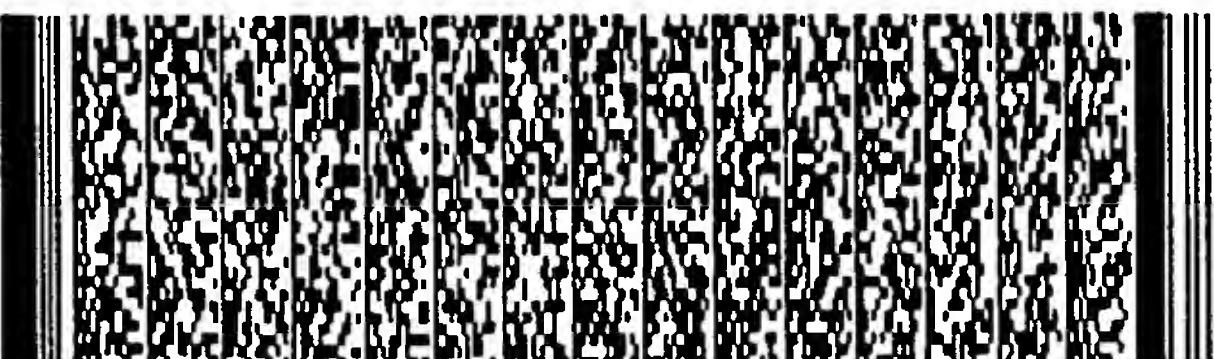
進行一裝置端檢查步驟，以便在將程式碼 77 實際寫入至儲存記憶體 60 之前，檢查程式碼 77 是否適用於周邊裝置 54。經由本發明主機 / 裝置端檢查步驟之實施，將可更進一步地確認韌體更新流程不會誤將不適用的韌體程式碼植入至周邊裝置。至於本發明主機 / 裝置端檢查步驟進行之方式，可以有數種實施例，以下將一一討論。

首先，請參考圖五 A（並一併參考圖三）。圖五 A 為本發明中主機 / 裝置端檢查步驟進行方式實施例之一的示意圖。如同前面所提過的，韌體研發廠商在韌體程式碼中都定義了一些關於韌體本身的資料，像是韌體研發廠商的名稱 (vendor ID)，該韌體程式碼適用之周邊裝置的型號名稱 (model name)，甚至於有序號 (serial number) 或者是版本資訊，形成一韌體辨識碼。即使在不同版本的韌體程式碼中，也都會統一記載有這些資訊。如圖五 A 所示，不論是周邊裝置 54 在韌體更新前既有的韌體程式碼 64，或者是韌體更新所使用的程式碼 76（或者是新式程式碼 77），只要是共同適用於周邊裝置 54 的韌體程式碼，就應該具有相同形式的韌體識別碼。一般來說，在韌體程式碼中，韌體識別碼之相關訊號是記錄於韌體程式碼中的一段常數 (constant) 中。如熟知技術者所知，韌體研發廠商都是以較高階的程式語言來將周邊裝置的各種操控程序編寫至一來源碼 (source code) 中，再經由一編譯器 (compiler) 軟體之編譯 (compile) 後，成為



五、發明說明 (22)

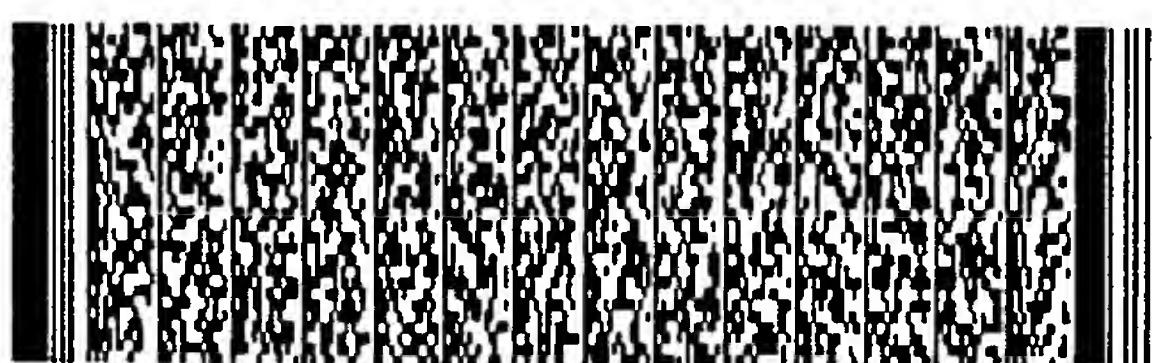
周邊裝置中控制電路可執行的二進位碼之韌體程式碼。韌體研發廠商通常在韌體程式碼的來源碼中以一常數陣列 (array) _pbTBLInquiry[]來編譯該韌體識別碼內容，該內容可以直接以數值表示，例如 0x05，或者以字元 (character) 編輯，例如 'A' 表示熟知技術者所知之 ASCII 碼，經編譯器編譯成為周邊裝置中控制電路可執行的二進位碼之韌體程式碼後，以一對應之二進位碼常數存在於該韌體程式碼中。故在韌體程式碼中，一定有部分內容是用來定義此一常數的內容值。舉例來說，如圖五 A 所示，在既有程式碼 64 之韌體別碼 64I 中，其部分內容就是以二進位碼方式來將常數 _pbTBLInquiry 之值定義為 {0x05, 0x80, …, 'A', 'b', …, 'd', 'M', …, 'k', 'm', …, …, …}，其中「Abcdefgh」就代表韌體研發廠商的名稱，「Model ikmh」就代表型號名稱，其他資料則可用來代表韌體版本等其他資料。同理，韌體更新所用的新程式碼 76 若為適用的韌體程式碼，其中一定也有一韌體識別碼 76I，同樣地用來定義常數 _pbTBLInquiry[] 之值。當要進行本發明之主機端檢查步驟時，主機 52 就可執行韌體更新之應用程式 74 (如圖三) 而搜尋新程式碼 76，以檢查新程式碼 76 中是否有部分內容是用來定義常數 _pbTBLInquiry 的，而此常數又被定義為特定的值。同時，主機 52 也可以要求周邊裝置 54 回傳既有程式碼 64 中其常數 _pbTBLInquiry 之值。當然，若新程式碼 76 中根本沒有定義此一常數，代表其一定不是正常的韌體程式



五、發明說明 (23)

碼。若新程式碼 76 中的確有定義此常數，且新程式碼 76 中此常數之值與既有程式碼 64 中定義之值相符（像是韌體研發廠商之名稱相同，型號名稱相同），主機 52 就可在主機端檢查步驟中判斷新程式碼 76 為適用的韌體程式碼。另外，在主機端檢查步驟中，主機 52 也可進一步地根據新程式碼 76 中的版本資訊來判斷新程式碼 76（相對於既有程式碼 64）是不是較新版本的韌體；若新程式碼 76 版本反而較舊，主機 52 也可判斷新程式碼 76 為不適用的韌體程式碼。進一步推廣此種比對的概念，本發明也可檢查新程式碼中某常數之值是否在某預設值的特定範圍內。像是在上述的例子中，就是檢查新程式碼中記載之版本是否大於既有程式碼之版本。

除了將新程式碼 76 中的韌體識別碼 76I 和既有程式碼 64 中的韌體識別碼 64I 比較以進行主機端檢查步驟外，韌體研發廠商也可在釋出韌體更新所用之應用程式 74 時，直接在應用程式 74 中記錄常數 _pbTBLInquiry 應有的格式及值（或是其值的合理範圍）；當主機 52 執行應用程式 74 而進行主機端檢查步驟時，就可以依據應用程式 74 中對常數 _pbTBLInquiry 的要求，直接判斷新程式碼 76 是否為適用的韌體程式碼，不必依據韌體識別碼 64I 中的資訊比對的基準。依據上述的道理，周邊裝置 54 中的控制電路 56 也可在進行裝置端檢查步驟時，利用既有程式碼 64 中對常數 _pbTBLInquiry 的定義，檢查暫存於緩衝記憶



五、發明說明 (24)

體 58 中的新程式碼 77 是否為適用的韌體程式碼。同理，在周邊裝置 54 出廠時，韌體製造廠商也可在控制電路 56 中預先設定常數 _pbTBLInquiry 應有之標準格式、內容值或內容值之合理範圍（像是版本編號應大於某一預設值），日後當控制電路 56 進行裝置端檢查步驟時，就可以獨立判斷新程式碼 77 是否具有正確的 _pbTBLInquiry 常數定義。在現行的資訊業界中，在韌體程式碼中加入韌體識別碼已經是一種標準的作法，而本發明也可方便地直接利用韌體識別碼來判斷韌體更新用的新韌體程式碼是否適用。

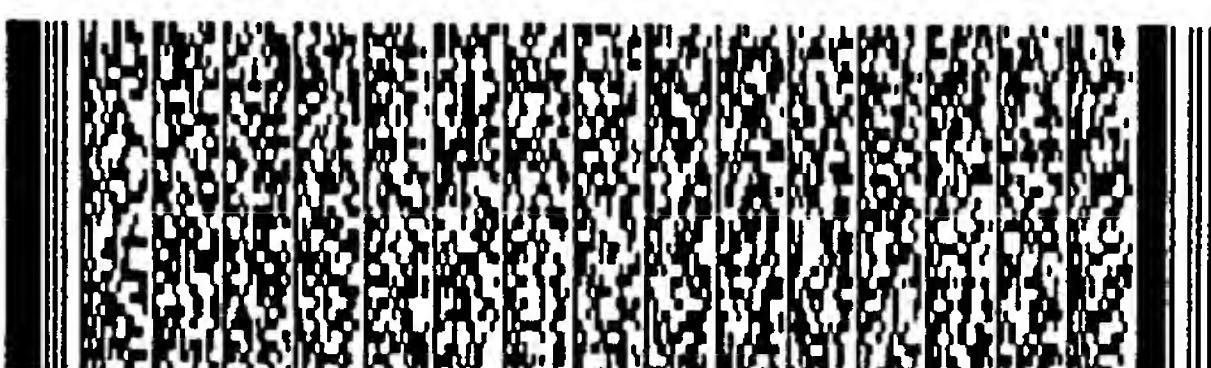
請繼續參考圖五 B(及圖三)。圖五 B 為本發明中主機 / 裝置端檢查步驟進行方式另一實施例之示意圖。除了利用韌體程式碼中普遍已定義之韌體識別碼來判斷韌體程式碼之適用性外，韌體研發廠商也可在韌體程式碼中預先插入特別定義的字串、數據，做為日後識別韌體程式碼適用性的依據。如熟知技術者所知，韌體研發廠商都是以較高階的程式語言來將周邊裝置的各種操控程序編寫至一來源碼 (source code) 中，再經由一編譯器 (complier) 軟體之編譯 (compile) 後，成為周邊裝置中控制電路可執行的二進位碼之韌體程式碼。韌體研發廠商可在韌體程式碼的來源碼中以一常數陣列 (array) 來編譯一特別定義之字串、數據，該內容可以直接以數值表示，例如 0x05，或者以字元 (character) 編輯，例如 'A'



五、發明說明 (25)

表示熟知技術者所知之 ASCII 碼，經編譯器編譯成為周邊裝置中控制電路可執行的二進位碼之韌體程式碼後，以一對應之二進位碼常數存在於該韌體程式碼中。如圖五 B 之實施例所示意的，韌體研發廠商可在韌體程式碼的來源碼 86 中額外加入程式片段 90A、90B，以分別定義出一字串 _pbSpecString(其內容為

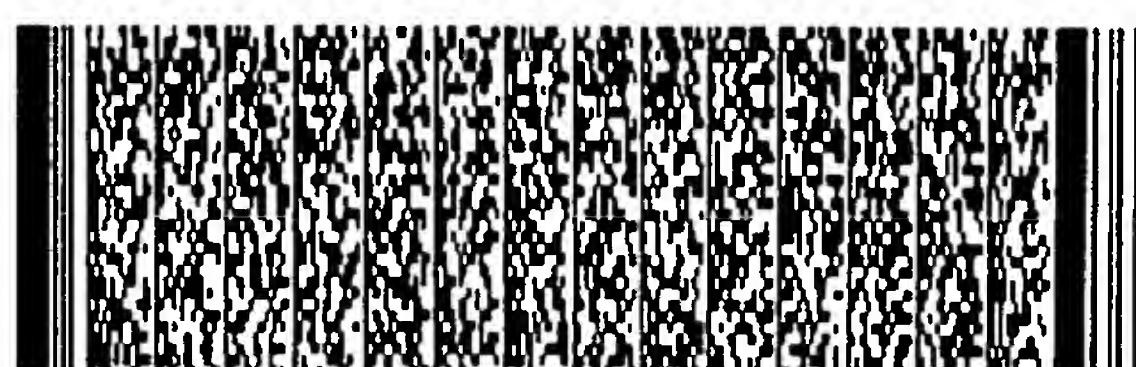
'M', 'e', 'd', 'i', 'a', 't', 'e', 'k') 以及一常數 _pbSpecValue。請注意，程式片段 90B 中不僅定義了常數 _pbSpecValue 之值，還以指令「_at_ 0xFFE0」指定了該常數值應存在一特定位址 0xFFE0(也就是十六進位的位址 FFE0)。根據此指令，編譯器就會在編譯過程中將此常數值置於該特定位址。等到來源碼 86 被編譯為二進位碼之韌體程式碼 88 後，對應於來源碼 86 中的程式片段 90A，韌體程式碼 88 中必定有部分內容 92A 是以二進位碼來記錄常數 _pbSpecString 之定義值；對應於程式片段 90B，由韌體程式碼 88 的位址 0xFFE0 處開始，也有部分內容 92B 是以二進位碼來記錄常數 _pbSpecValue 的定義值。韌體程式碼 88 釋出後，也就成為韌體更新用的新程式碼（像是圖三中的新程式碼 76）。利用新程式碼中特別定義的字串、常數，就可以實施本發明中的主機 / 裝置端檢查步驟。舉例來說，主機 52、控制電路 56 在進行主機、裝置端檢查步驟時，可分別檢查新程式碼 76、77 中是否記錄有「Mediatek」這個字串以定義字串 _pbSpecString。因為韌體研發廠商釋出的適用新程式碼必定有部分內容是



五、發明說明 (26)

用來記錄「Mediatek」以定義字串 _pbSpecString，若在主機 / 裝置端檢查步驟中發現新程式碼並沒有記錄「Mediatek」，就代表該新程式碼為不適用的程式碼。同理，主機 52、控制電路 56也可在主機 / 裝置端檢查步驟中檢查新程式碼是否在位址 0xFFE0處存有常數 _pbSpecValue正確的定義值。若新程式碼中在位址 0xFFE0記錄的不是常數 _pbSpecValue的正確定義值，就代表新程式並非適用的韌體程式碼。或者，也可在進行主機 / 裝置端檢查步驟時，檢查常數 _pbSpecValue的定義值是否存在新程式碼特定位址 0xFFE0處。當然，要以上述方式來實施本發明主機 / 裝置端檢查步驟時，韌體製造商在周邊裝置 54出廠前，就要預先設定控制電路 56（圖三），以使控制電路 56在日後進行裝置端檢查步驟時，能知道要比對的目標（也就是圖三中的預設內容 82；像是要在新程式碼中尋找「Mediatek」，或是在新程式碼的那個特定位址應該存有哪些特定值）。同理，韌體製造廠商也要在釋出韌體更新之應用程式 74前，預先在應用程式 74中記錄主機端裝置檢查步驟要比對的目標，讓主機 52能在執行應用程式 74後依照上述的比對原理進行主機端檢查步驟。

請參考圖五 C。圖五 C為本發明主機 / 裝置端檢查步驟進行方式另一實施例的示意圖。除了上述以韌體程式碼中特別定義之常數、定值字串來進行本發明之檢查步驟



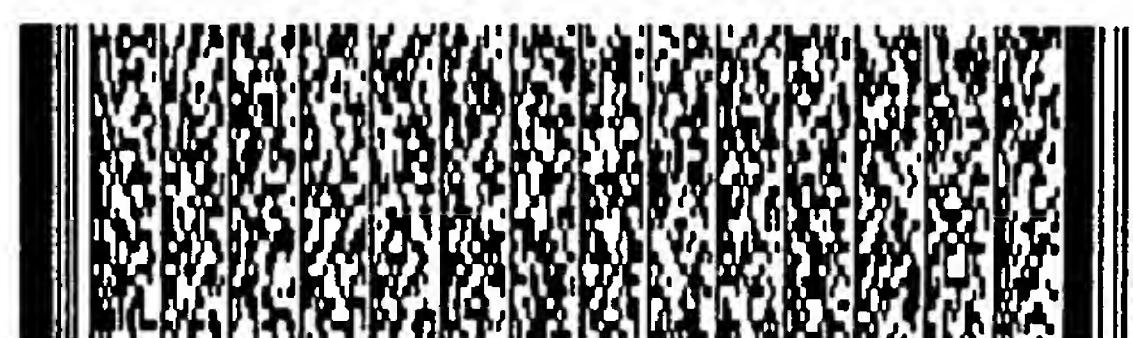
五、發明說明 (27)

外，也可以根據韌體程式碼中特定的指令來進行本發明之檢查步驟。如圖五 C所示，韌體研發廠商在編寫韌體程式碼的來源碼 86時，可加入程式片段 90C，以「 CSEG AT FF80H」之指令將一段預設之指令碼 94編譯至位址 FF80H（也就是十六進位的位址 FF80）。指令碼 94中定義的可以是有意義、有實際用途的操控程序，或是一些並無實際用途的冗餘操作（像是將兩變數之值交換後再換回來）。圖五 C中示意的指令碼 94，其第一行「 MOV DRTP, #0800H」即用來讓一指標 (pointer)DRTP指向緩衝記憶體中的一個位址 0800H(十六進位的位址 0800)；第二行「 MOVX A,@ DRTP」則將一暫存器 A中儲存之值移到指標 DRTP指向的位址，等等。在將來源碼 86編譯為韌體程式碼 88後，根據程式片段 90C所指定的位址 FF80H，韌體程式碼 88就會在起始於位址 FF80H的部分內容 92C中，以二進位碼之形式來記錄指令碼 94。將韌體程式碼 88釋出為韌體更新所用之新程式碼後，就可依據新程式碼是否在位址 FF80H有對應指令碼 94的部分內容 92C(或是搜尋部分內容 92C是否起始於位址 FF80H)，來進行本發明之主機 / 裝置端檢查步驟，以判斷新程式碼是否為韌體研發廠商所釋出的適用韌體程式碼。類似於圖五 B中實施例之情況，應用程式 74、控制電路 56也要在事先設定好要比對目標（像是指定位址、指令碼 94對應之二進位碼為何），方便日後進行主機 / 裝置端檢查步驟時，做為檢查進行的依據。



請參考圖五 D。圖五 D為本發明主機 / 裝置端檢查步驟又一實施例之示意圖。如圖五 D所示，韌體研發廠商也可在韌體程式碼的來源碼 86中加入程式片段 90D，以便在編譯後於韌體程式碼 88中之特定位址加入特定之值。舉例來說，圖五 D中的程式片段 90D，指令「 CSEG AT 0005H」及下一行的「 DB E1H」就是指定要將一位元組的資料（內容為十六進位的 E1）記錄於韌體程式碼 88中的位址 0005H(十六進位之位址 0005)；指令「 CSEG AT FFFEH」及次行的「 DB E2H」則將一位元組之值 E2記錄於位址 FFFEH。將來源碼 86編譯為韌體程式碼 88後，韌體程式碼 88就會在位址 0005H的部分內容 92D1處以二進位碼記錄有十六進位的數值 E1，在位址 FFFEH之部分內容 92D2處記錄有十六進位的數值 E2。在進行本發明之主機 / 裝置端檢查步驟時，即可檢查新程式碼在特定位址處是否記錄有特定之值（像是在位址 0005H處是否為數值 E1），以判斷新程式碼是否為韌體研發廠商釋出的適用韌體程式碼。

請參考圖五 E。圖五 E為本發明主機 / 裝置端檢查步驟進行方式另一實施例之示意圖。除了在來源碼中以程式片段插入特定的資料、指令以做為判斷新程式碼是否適用的依據外，本發明也可在編譯後的韌體程式碼中插入特定的資料，做為檢查步驟進行的依據。如圖五 E所示，



五、發明說明 (29)

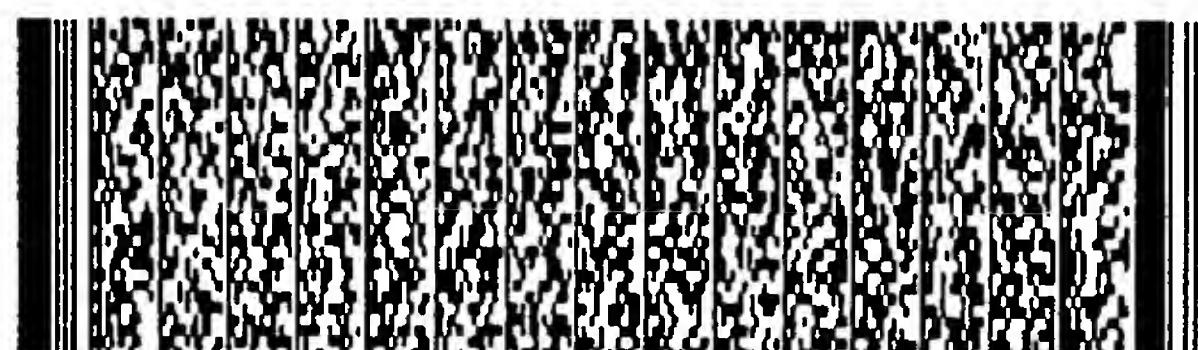
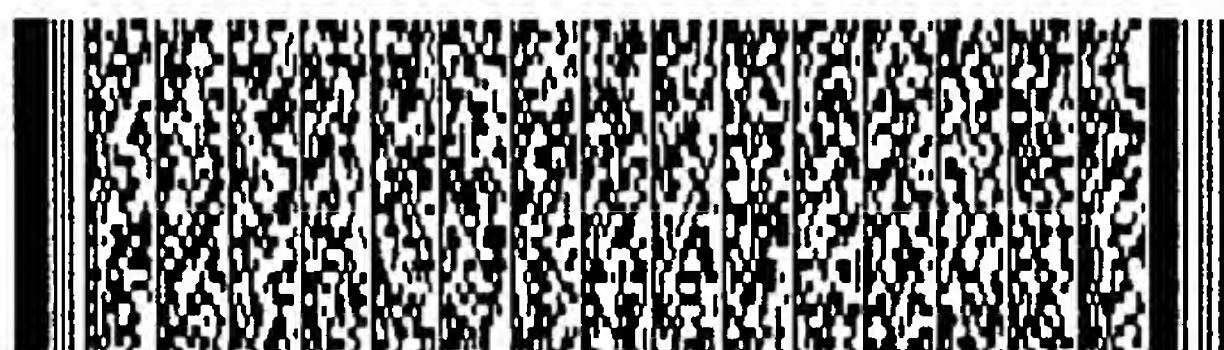
一般來說，當韌體程式碼之來源碼 86 經過編譯而成為韌體程式碼 88 後，韌體程式碼 88 除了用來記錄指令、數據（像是常數值）的部分內容外，還會留下一些未使用區段（unused segment）。這些未使用區段會被填入特定的填充資料。像在圖五 E 中，在部分內容 92E1 到 92E4 標記為斜線的部分，都記錄了有意義之程式、指令所對應的二進位碼，是所謂的程式碼區段（code segment）；被填入十六進位「F」的部分，即是所謂的未使用區段，沒有用來記錄任何程式、指令。像在部分內容 92E2、92E3 之間，就有未使用區段。另外，由於韌體程式碼 88 多半會被編譯為固定大小（像是 512Kbyte）的程式碼，以方便最後使用時，會在各個程式區段間跳躍地讀取指令，不會執行到未使用區段，故本發明也可在韌體程式碼 88 中的未使用區段插入特定的資料做為標記，不僅不影響周邊裝置執行韌體程式碼的運作，而插入的標記也可當作本發明主機端 / 裝置端檢查步驟進行時檢查的依據。如圖五 E 所示，在韌體程式碼 88 中的未使用區段插入數筆資料 95 後，就變成了韌體程式碼 89；而韌體程式碼 89 也就是韌體研發廠商正式釋出的新程式碼。當要進行本發明之主機 / 裝置端檢查步驟時，就可在新程式碼之未使用區段中的特定位址搜尋是否有插入的資料，判斷新程式碼是否



五、發明說明 (30)

適用。類似於本發明於圖五 B、C、D 中的實施例，韌體研發廠商也要事先設定韌體更新之應用程式 74 及控制電路 56，使主機 52、周邊裝置 54 在進行檢查步驟時，能知道比對的目標，像是在未使用區端中的那個位址應該會有什麼樣的資料。相對於圖五 E 之實施例是將標記之資料插入至未使用區段，在圖五 A 至圖五 D 之實施例中，用來當作主機端 / 裝置端檢查步驟進行依據之資料都會位於程式碼區段。

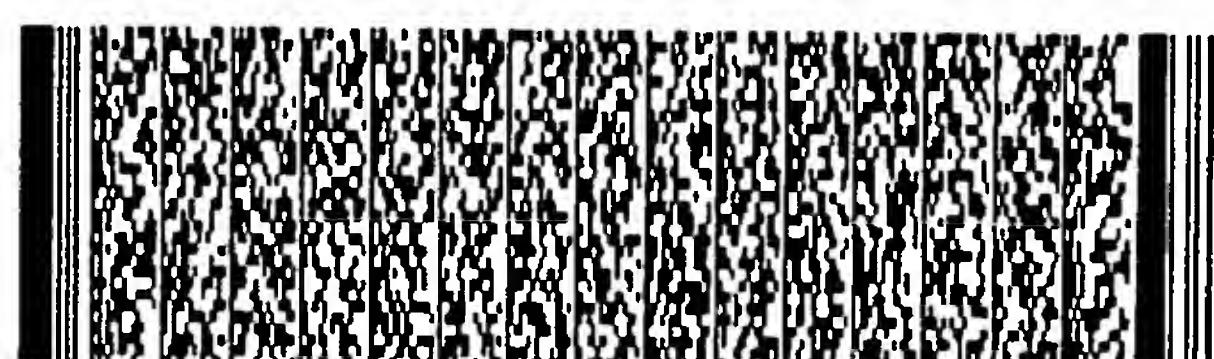
除了上述的方法，本發明主機端 / 裝置端檢查步驟尚有以下數種其他的工作法。舉例來說，在進行檢查步驟時，可搜尋新程式碼中某一特定的指示資料（像是字串、常數）所在位址，由該位址平移一段預設之平移位址，再檢查新程式碼於該新位址是否符合另一預設內容。換句話說，當韌體研發廠商在釋出韌體程式碼時，不但要在程式碼中加入指示資料，也要在該指示資料所在位址平移該平移位址的地方，加入該預設內容，以在程式碼釋出後，配合客戶進行主機端 / 裝置端檢查步驟。另外，在進行檢查步驟時，也可檢查新程式碼中位於不同預設位址的不同常數在經過預設之運算後，是否符合一預設值。舉例來說，檢查新程式碼中分別位於兩不同位址之常數相加後，是否為一預定值。在前面曾經提到過，檢查步驟可檢查韌體識別碼中韌體版本的數值大小做為韌體程式碼適用



五、發明說明 (31)

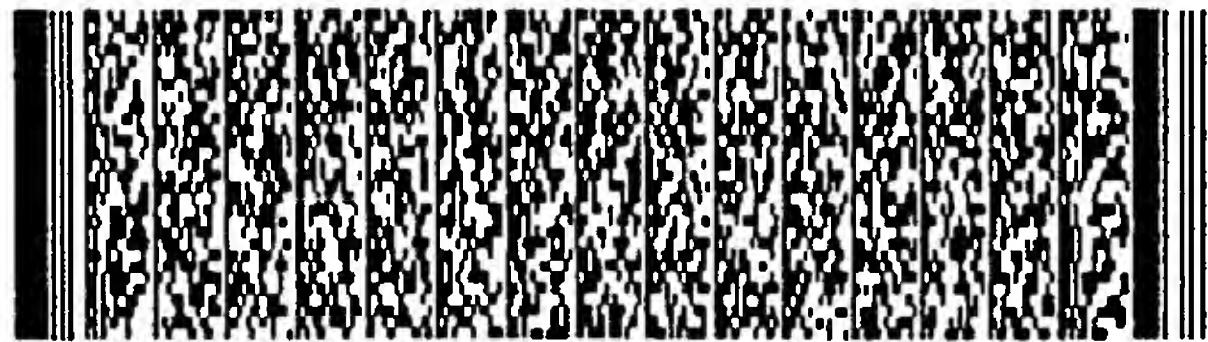
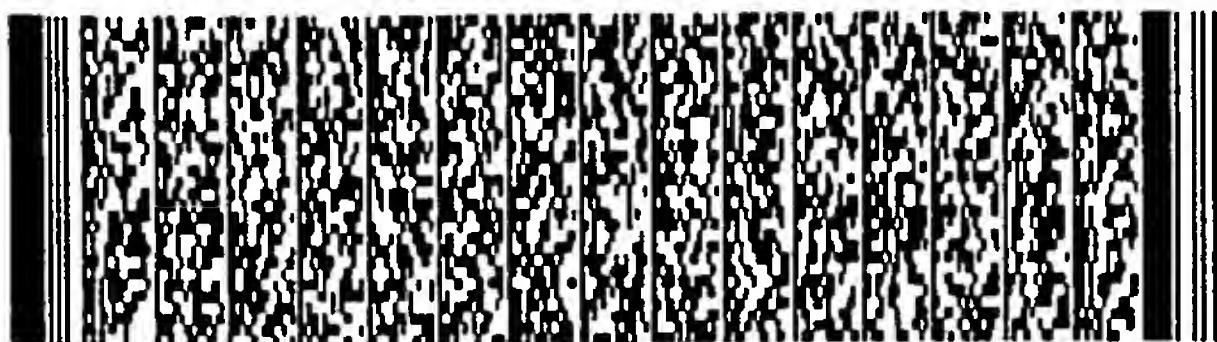
於否之依據，不過惡意的破壞者也可能在竄改程式碼時，一併竄改韌體版本的數值，意圖躲過檢查步驟的檢查。而在韌體更新過程中將竄改後的韌體程式碼植入周邊裝置中。針對這種情形，韌體研發廠商就可在用常數，而預設程式碼中，於另一預設之位址植入另一版本數值相加為更大之數值；換句話說，在更新、韌體版本數值相加後為更大的數值。而當進行主機端 / 裝置檢查時，主機 / 周邊裝置除了檢查韌體版本之數值是否大於此檢查用常數也隨之變小。這樣一來，就能進一步確認韌體版本數值本身是否正確。

由上述討論延伸，本發明主機端 / 裝置端檢查步驟也可以先搜尋兩個預設內容資料分別所在的兩個位址，再檢查這兩個位址之間的程式碼是否具有適用之特性。舉例來說，韌體研發廠商在釋出的串連數據預設的資料在經過一連串的依循規律遞增或遞減；或者是在兩預設內容資料之間的資料是預設內容資料相加後的值為一定值（或是兩預設內容資料相減）；或者是在兩預設內容資料相加後，能得到一預設之標準值。這樣一來，當進行主機端 / 裝置端檢查步驟時，主機 / 周邊裝置就能檢查在新程式碼中，此兩預設內容資料間的資料是



五、發明說明 (32)

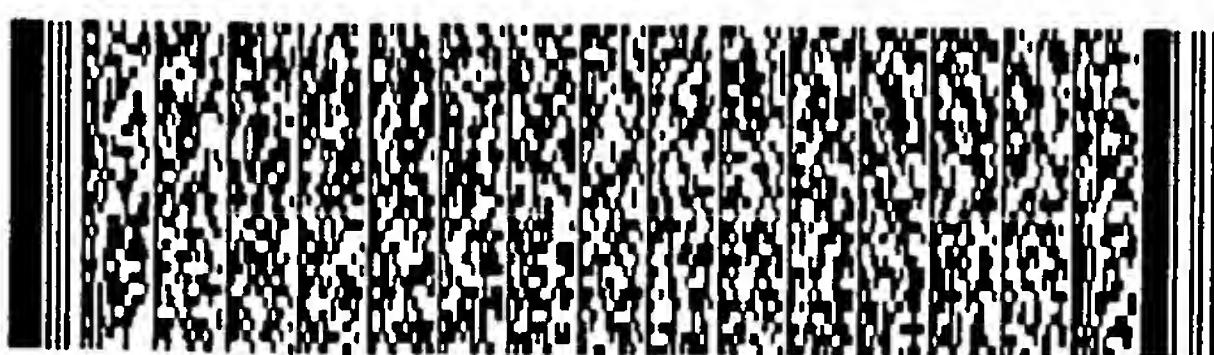
否合乎預設之規律，或是否能在以預設演算方法處理廠商可在
到預設之標準值。在這種作法之下，勒體研發資料間加入能得到標
不同的版本的勒體程式碼中，在兩預設內容資料法後均能比對標
同一的資料，但加入的資料在經過預設演算程式碼中，兩分析
同一暴露於程式碼中。因為不同版本的適用程式碼故意破壞者分
預設內容資料間的資料均會相異，即使惡意破壞特定的規律避
不同版本的適用程式碼，也難以歸納出特定的規律避
開本發明檢查步驟的篩檢。



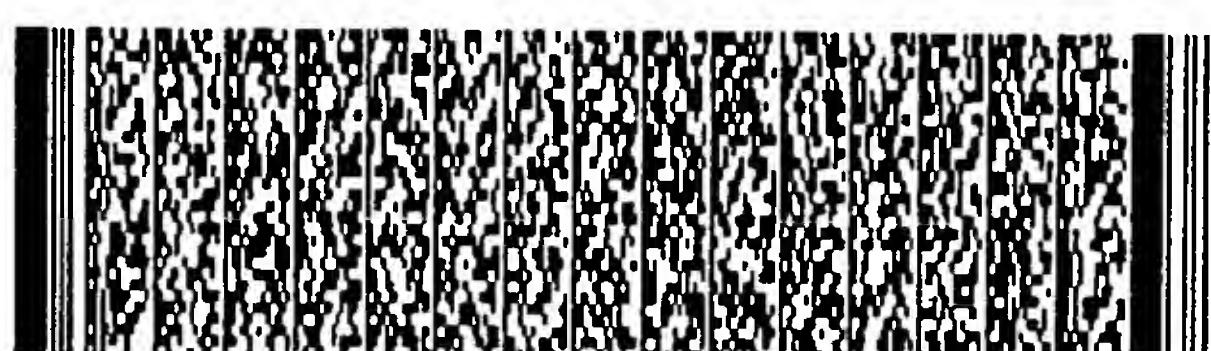
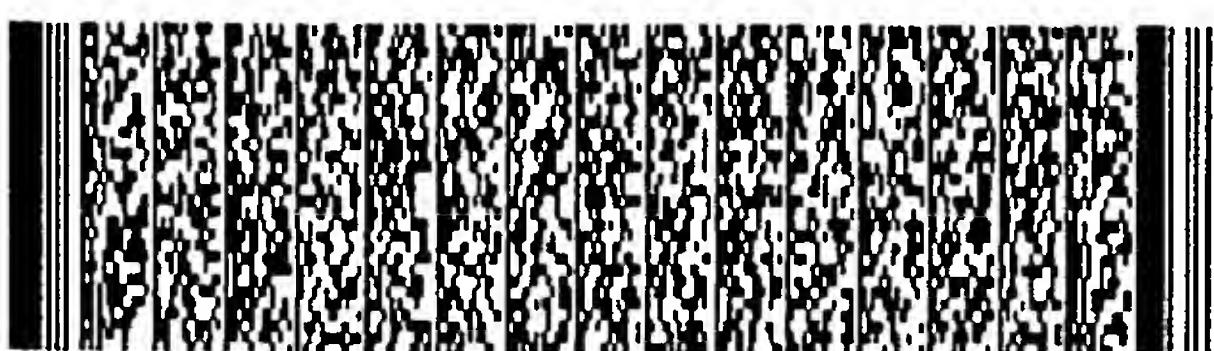
五、發明說明 (33)

的韌體程式碼不會在韌體更新流程中被誤植入周邊裝置中。在設定周邊裝置 54之控制電路 56的比對目標時，由於控制電路 56是依據既有程式碼來運作的，所以比對的目標、裝置端檢查步驟進行的方式，也可以記錄於周邊裝置 54預建之韌體程式碼中。在實施本發明時，圖五 A至圖五 E的各種實施方法能單獨使用，或是合併多種方法一起實施，而主機端 / 裝置端檢查步驟也可使用不同的比對目標。舉例來說，主機 52可使用圖五 B中的實施例，檢查新程式碼中是否有特定字串以實施主機端檢查步驟，而控制電路 56則能利用圖五 A中的實施例，以韌體識別碼來為依據來進行裝置端檢查步驟，判斷新程式碼是否符合。當然，在此情況下，適用的韌體程式碼中不但要有韌體識別碼以支援圖五 A中實施例之實施，還要有特定字串以支援圖五 B中實施例之實施。正由於本發明主機端 / 裝置端檢查步驟檢查之依據是由韌體研發廠商所預設的，故可提供一準確的比對目標，確實地檢查出不適用的韌體程式碼。相較之下，圖二中的習知流程 100雖會根據韌體更新用之新程式碼，利用檢查碼產生演算法算出一檢查碼，但由於習知流程根本無從得知適用韌體程式碼對應的檢查碼，故無法利用檢查碼檢查出不適用的新程式碼。

雖然本發明示於圖四之流程 200在主機、裝置端分別進行了主機端、裝置端檢查步驟，但本發明也可只由主



五、發明說明 (34)



五、發明說明 (35)

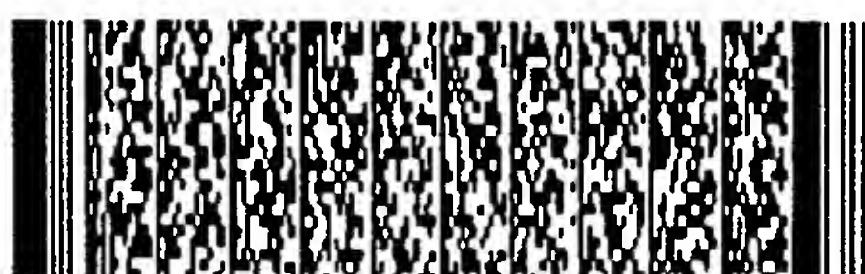
運作之裝置不會在韌體更新的過程中被植入不適用的韌體程式碼；尤其是在裝置中由裝置本身自行進行的裝置端檢查步驟，可主動確保裝置本身不會被誤植入不適用的韌體程式碼。

在習知的韌體更新流程中，無法針對韌體更新所用之程式碼進行內容的檢查，故也無法在韌體更新的裝置中避免不適用的新程式碼被錯誤地植入。在韌體更新的流程中，相較之下，本發明在韌體更新的流程中加入了主機/裝置端檢查步驟，能依據新程式碼的不同周邊裝置間韌體邊界來判斷新邊緣裝置是否適用。另外，由於本發明可用主機/裝置端檢查步驟確認韌體更新是否適用，就可以在主機端進行該式建傳來周邊裝置單一整體之更新。在主機裝置端檢查步驟，並確認不同周邊裝置間韌體邊界是否適用，就可程式取得已確認能使用同一韌體更新，讓使用者進行韌體更新時更為簡易、方便。



五、發明說明 (36)

以上所述僅為本發明之較佳實施例，凡依本發明申請專利範圍所做之均等變化與修飾，皆應屬本發明專利之涵蓋範圍。



圖式簡單說明

圖式之簡單說明：

圖一為一典型電子系統中主機及周邊裝置配置之示意圖。

圖二為圖一中電子系統以一習知流程進行韌體更新之流程示意圖。

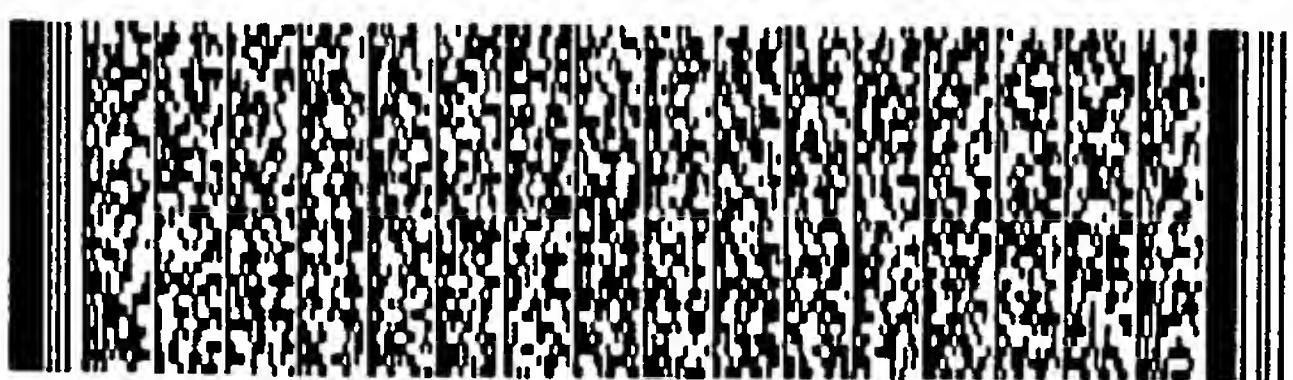
圖三為本發明電子系統中主機及周邊裝置配置之示意圖。

圖四為圖三中電子系統以本發明流程進行韌體更新之流程示意圖。

圖五A至五E分別為圖四中主機端/裝置端檢查步驟不同實施例之示意圖。

圖式之符號說明：

10、50	電子系統	12、52	主機
14-15、54-55			周邊裝置
16、56	控制電路	18、58	緩衝記憶體
20、60	儲存記憶體	22、62	伺服硬體
24、64	程式碼		
24I、64I、76I、77I			韌體識別碼
26、66	中央處理器		
28A、68A	北橋電路	28B、68B	南橋電路
30、70	記憶體	32A、72A	繪圖卡



圖式簡單說明

32B、72B 顯示器	34、74	應用程式
36-37、76-77、84		新程式碼
36C、37C、39C、76C、77C、79C		檢查碼
56B 檢查模組		
80、82 預設內容	86	來源碼
88、89 勅體程式碼	90A-90D	程式片段
92A-92C、92D1、92D2		部分內容
94 指令碼	95	資料
100、200 流程		
102-122、202-224		步驟



六、申請專利範圍

1. 一種於一電子系統中更新程式碼的方法，其中該電子系統包含有一主機以及一周邊裝置；

該周邊裝置包含有：

一控制電路，用來執行一第一程式碼，以根據該主機傳來的控制指令控制該周邊裝置的運作；

而該方法包含有：

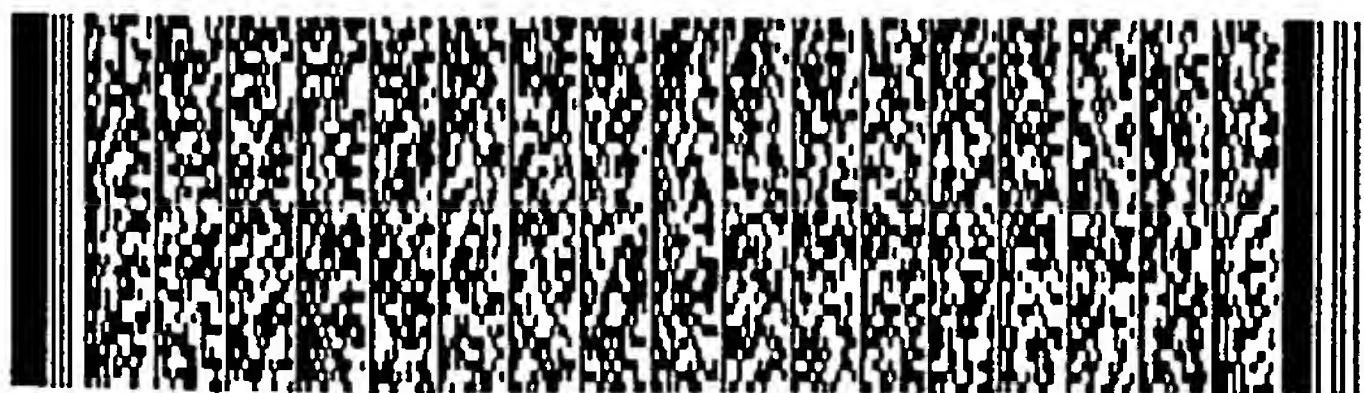
取得一第二程式碼；以及

在以該第二程式碼取代該周邊裝置之第一程式碼之前，進行一主機端檢查步驟，以利用該主機檢查該第二程式碼的部分內容是否符合一預設內容。

2. 如申請專利範圍第1項之方法，其中該周邊裝置另包含有一儲存記憶體，用來以非揮發性的方式儲存該第一程式碼；而當要以該第二程式碼取代該第一程式碼時，係將該第一程式碼由該儲存記憶體抹除，並將該第二程式碼寫入至該儲存記憶體中。

3. 如申請專利範圍第1項之方法，其中當在以一第二程式碼取代該第一程式碼之前而進行該主機端檢查步驟時，係在該控制電路執行該第二程式碼以控制該周邊裝置的運作前，進行該主機端檢查步驟。

4. 如申請專利範圍第1項之方法，其中該預設內容係該第一程式碼的部分內容，或是記錄於該第一程式碼中的



六、申請專利範圍

一常數；而當進行該主機端檢查步驟時，係檢查該第二程式碼中是否具有該第一程式碼的部分內容，或檢查該第二程式碼中記錄之常數之值是否相當於該第一程式碼中該常數之值，或檢查該第二程式碼中記錄之常數之值是否在該第一程式碼中該常數之值的預設範圍內。

5. 如申請專利範圍第1項之方法，其中該預設內容係為固定之內容，使得當該第二程式碼改變時，該預設內容亦不會改變。

6. 如申請專利範圍第1項之方法，其中當進行該主機端檢查步驟時，係由該主機讀取該第二程式碼中位於預設位址的部分內容，以檢查該部分內容是否符合該預設內容，或搜尋該第二程式碼中是否存在該預設內容。

7. 如申請專利範圍第1項之方法，其另包含有：在進行該主機端檢查步驟後，若該第二程式碼的部分內容不符合該預設內容，則停止以該第二程式碼取代該第一程式碼。

8. 如申請專利範圍第1項之方法，其另包含有：在進行該主機端檢查步驟後，若該第二程式碼的部分內容符合該預設內容，則以該第二程式碼取代該第一程式碼，使該控制電路得以執行該第二程式碼以控制該周邊



六、申請專利範圍
裝置的運作。

9. 一種於一電子系統中更新程式碼的方法，其中該電子系統包含有一主機以及一周邊裝置；

該周邊裝置包含有：

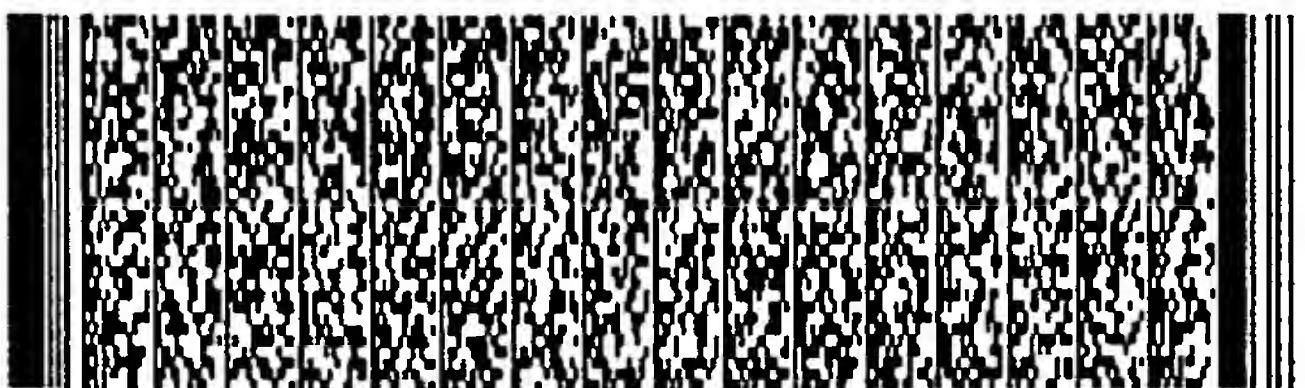
一控制電路，用來執行一第一程式碼，以控制該周邊裝置的運作；

而該方法包含有：

將一第二程式碼由該主機傳輸至該周邊裝置；以及在以該第二程式碼取代該第一程式碼之前，進行一裝置端檢查步驟，以利用該控制電路檢查該第二程式碼的部分內容是否符合一預設內容。

10. 如申請專利範圍第9項之方法，其中該周邊裝置另包含有一非揮發性的儲存記憶體，用來以非揮發性的方式儲存該第一程式碼；而當要以該第二程式碼取代該第一程式碼時，係將該第一程式碼由該儲存記憶體抹除，並將該第二程式碼寫入至該儲存記憶體中。

11. 如申請專利範圍第9項之方法，其中當在以一第二程式碼取代該第一程式碼之前而進行該裝置端檢查步驟，係在該控制電路執行該第二程式碼以控制該周邊裝置的運作前，進行該裝置端檢查步驟。



六、申請專利範圍

12. 如申請專利範圍第9項之方法，其中該預設內容係該第一程式碼的部分內容，或是記錄於該第一程式碼中的一常數；而當進行該裝置端檢查步驟時，係檢查該第二程式碼中是否具有該第一程式碼的部分內容，或檢查該第二程式碼中該常數之值是否相當於該第一程式碼中該常數之值，或檢查該第二程式碼中記錄之常數之值是否在該第一程式碼中該常數之值的預設範圍內。

13. 如申請專利範圍第9項之方法，其中該預設內容係為固定之內容，使得當該第二程式碼改變時，該預設內容亦不會改變。

14. 如申請專利範圍第9項之方法，其中當進行該裝置端檢查步驟時，係由該控制電路讀取該第二程式碼中位於預設位址的部分內容，以檢查該部分內容是否符合一預設內容，或搜尋該第二程式碼中是否存該此預設內容。

15. 如申請專利範圍第9項之方法，其另包含有：在進行該裝置端檢查步驟後，若該第二程式碼的部分內容不符合該預設內容，則停止以該第二程式碼取代該第一程式碼。

16. 如申請專利範圍第9項之方法，其另包含有：在進行該裝置端檢查步驟後，若該第二程式碼的部分內



六、申請專利範圍

容符合該預設內容，則以該第二程式碼取代該第一程式碼，使該控制電路得以執行該第二程式碼以控制該周邊裝置的運作。

17. 如申請專利範圍第9項之方法，其中該周邊裝置另包含有一緩衝記憶體，用來以揮發性的方式儲存資料；而當進行該裝置端檢查步驟時，該控制電路係將該第二程式碼暫存於該緩衝記憶體中，以讀取該第二程式碼的部分內容而進行該裝置端檢查步驟。
18. 如申請專利範圍第17項之方法，其中該周邊裝置另包含有一非揮發性的儲存記憶體，用來以非揮發性的方式儲存該第一程式碼；而當要以該第二程式碼取代該第一程式碼前而進行該裝置端檢查步驟時，係在該第一程式碼尚未被抹除而該第二程式碼尚未被寫入至該儲存記憶體前，進行該裝置端檢查步驟。
19. 如申請專利範圍第9項之方法，其中該周邊裝置為一光碟機。
20. 一周邊裝置，其包含有：
 - 一控制電路，用來執行一第一程式碼，以控制該周邊裝置的運作；
 - 而該控制電路中設有一檢查模組，該檢查模組可在



六、申請專利範圍

該控制電路以一第二程式碼取代該第一程式碼之前，檢查該第二程式碼的部分內容是否符合一預設內容。

21. 如申請專利範圍第20項之周邊裝置，其另包含有一該非揮發性的儲存記憶體，用來以該第二程式碼取代該第一程式碼；而當該控制電路以該第二程式碼由該儲存記憶體抹除，並將該第二程式碼寫入至該儲存記憶體中。
22. 如申請專利範圍第20項之周邊裝置，其中當該控制電路以一第二程式碼取代該第一程式碼之前而由該檢查模組進行檢查時，係在該控制電路執行該第二程式碼以控制該周邊裝置的運作前，由該檢查模組進行檢查。
23. 如申請專利範圍第20項之周邊裝置，其中該預設內容係該第一程式碼的部分內容，或是記錄於該第一檢查模組進行檢查時，係該第一程式碼的部分內容，或第一常數碼中的一常數；而當該檢查模組進行檢查時，係該第二程式碼中是否具有該第一程式碼的第一常數之值，或檢查該第二程式碼中該常數之值，或檢查該第一程式碼中該常數之值的預設範圍內。
24. 如申請專利範圍第20項之周邊裝置，其中該預設內容不會隨該第二程式碼改變而改變。



六、申請專利範圍

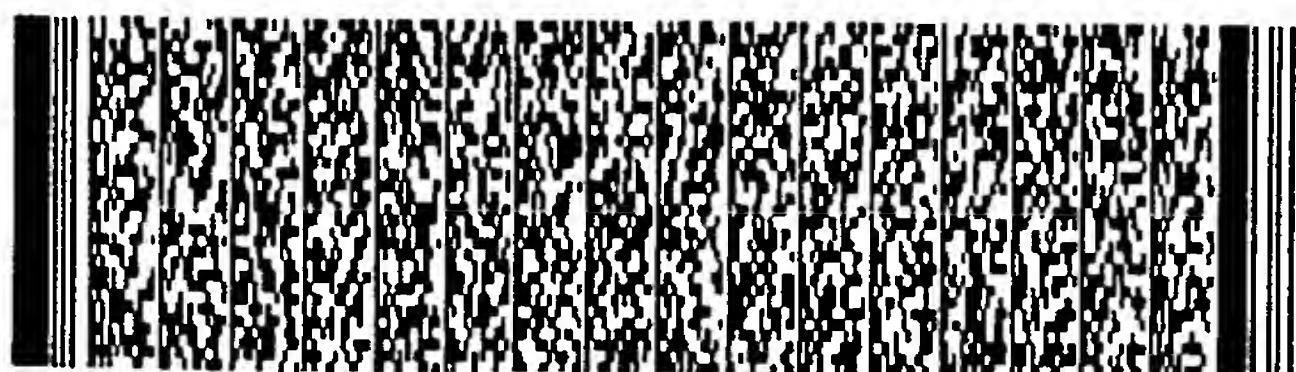
25. 如申請專利範圍第 20 項之周邊裝置，其中當該檢查模組進行檢查時，係由該控制電路讀取該第二程式碼中位於預設位址的部分內容，以便由該檢查模組檢查該部分內容是否符合一預設內容，或搜尋該第二程式碼中是否存在此預設內容。

26. 如申請專利範圍第 20 項之周邊裝置，其中若該檢查模組檢查出該第二程式碼的部分內容不符合該預設內容，則該控制電路會停止以該第二程式碼取代該第一程式碼。

27. 如申請專利範圍第 20 項之周邊裝置，其中若該檢查模組檢查出該第二程式碼的部分內容符合該預設內容，則該控制電路會以該第二程式碼取代該第一程式碼，使該控制電路得以執行該第二程式碼以控制該周邊裝置的運作。

28. 如申請專利範圍第 20 項之周邊裝置，其另包含有一緩衝記憶體，用來以揮發性的方式儲存資料；而該控制電路係將該第二程式碼暫存於該緩衝記憶體中，並在讀取該第二程式碼的部分內容後由該檢查模組進行檢查。

29. 如申請專利範圍第 28 項之周邊裝置，其係使用於一



六、申請專利範圍

電子系統，該電子系統中另設有一主機，而該第二程式碼係由該主機傳輸至該周邊裝置。

30. 如申請專利範圍第 28 項之周邊裝置，其另包含有一該非揮發性的儲存記憶體，用來以非揮發性的方式儲存該第一程式碼；而當該控制電路要以該第二程式碼取代該第一程式碼前，而由該檢查模組進行檢查時，係在該第一程式碼尚未被抹除而該第二程式碼尚未被寫入至該儲存記憶體前，由該檢查模組進行檢查。

31. 一種於一電子系統中更新程式碼的方法，其中該電子系統包含有一主機以及一周邊裝置；

該周邊裝置包含有：

一控制電路，用來執行一第一程式碼，以根據該主機傳來的控制指令控制該周邊裝置的運作；

而該方法包含有：

取得一第二程式碼；以及

在以該第二程式碼取代該周邊裝置之第一程式碼之前，進行一檢查步驟，以根據該第二程式碼的內容求出一對應的內容特性，並檢查該內容特性是否符合一預設特性；

而該預設特性不會隨該第二程式碼之內容改變而改變。

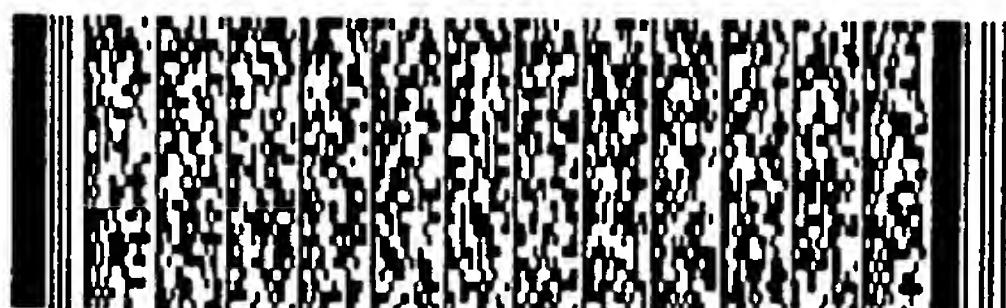


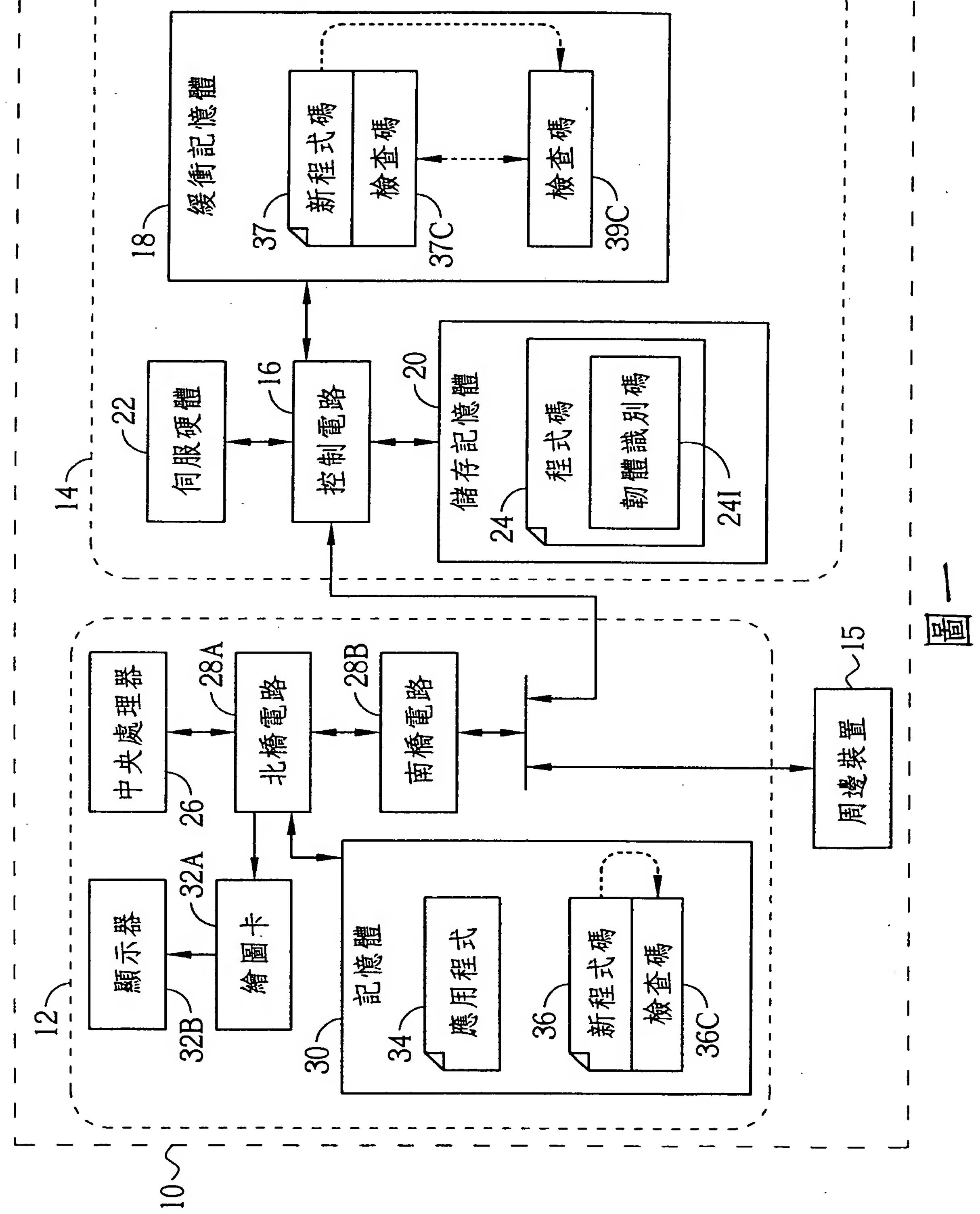
六、申請專利範圍

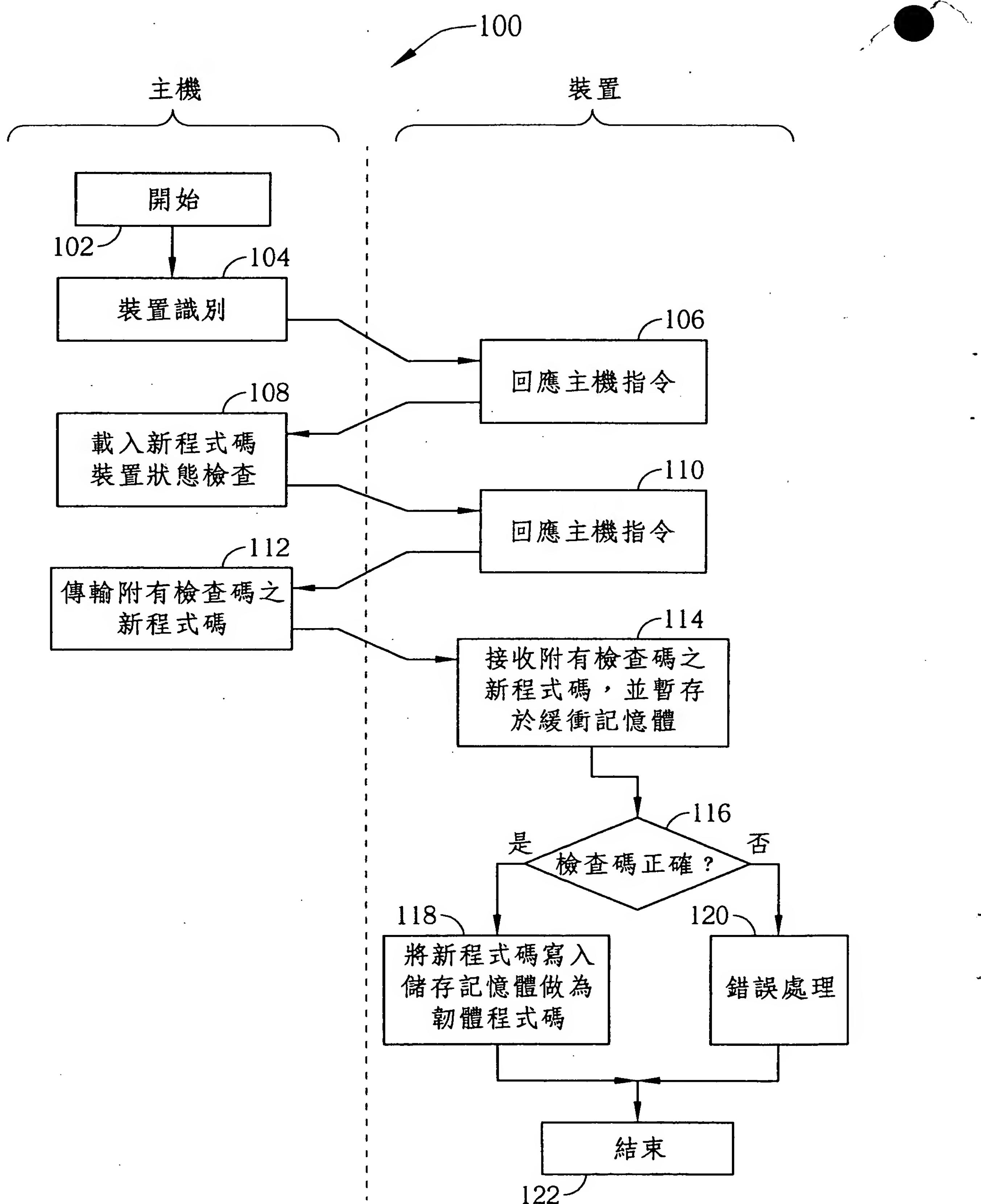
32. 如申請專利範圍第31項之方法，其中該檢查步驟係由該主機來進行。

33. 如申請專利範圍第31項之方法，其中該檢查步驟係由該控制電路來進行。

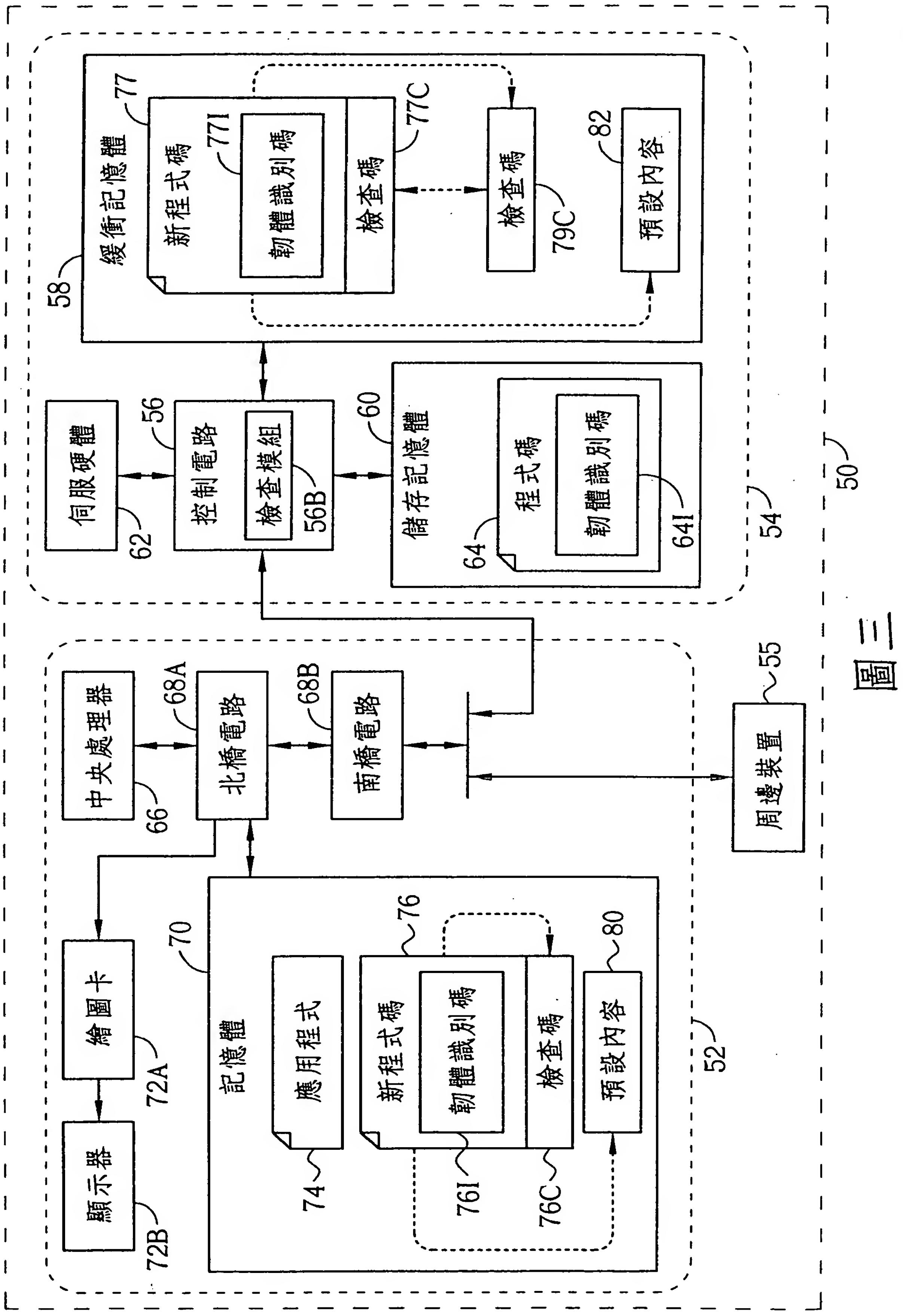
34. 如申請專利範圍第31項之方法，其中該內容特性係一預設內容於該第二程式碼中所在之位址，而該預設特性係一預設位址；而當進行該檢查步驟時，係檢查該第二程式碼中該預設內容所在的位址是否符合該預設位址。



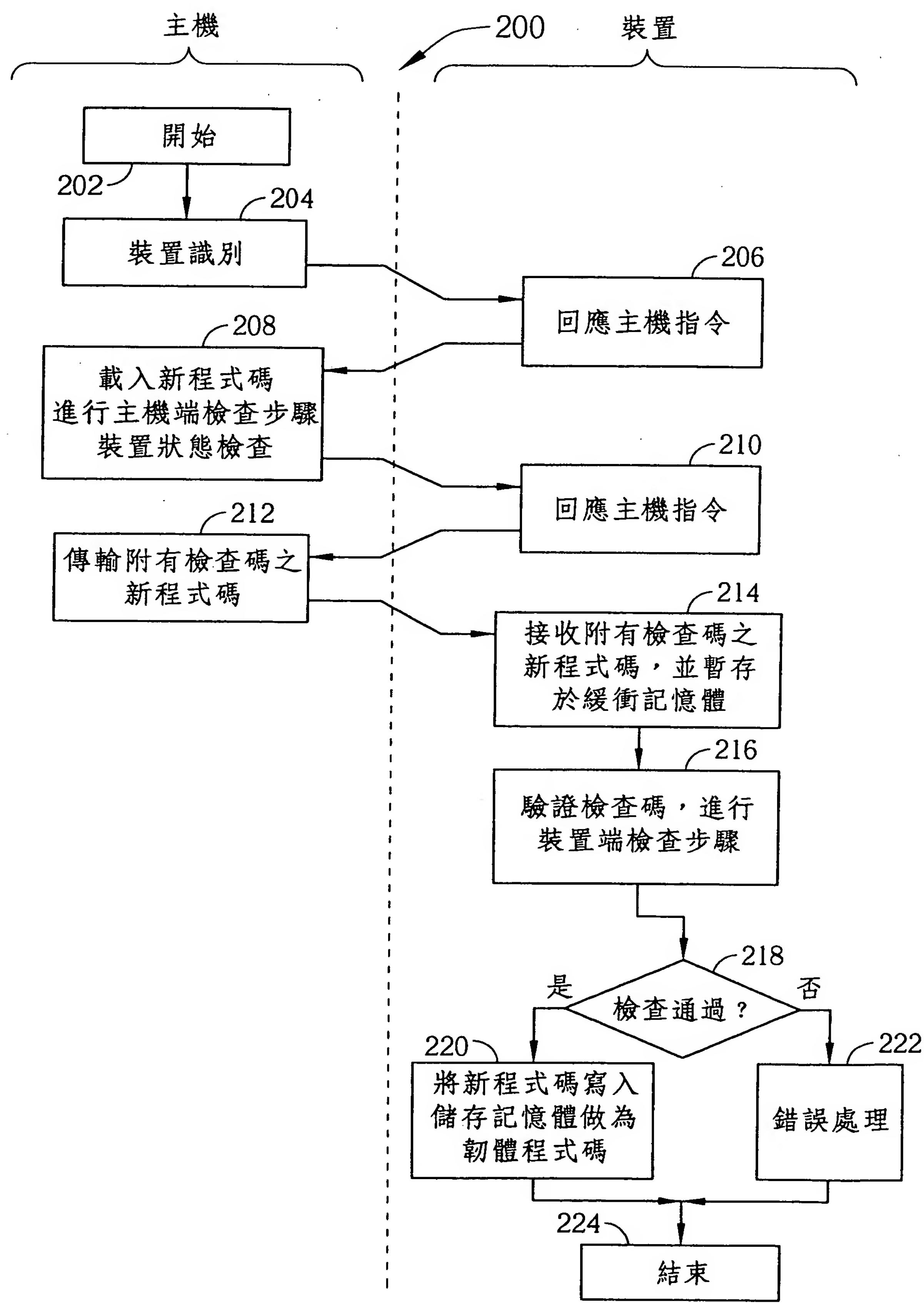




圖二

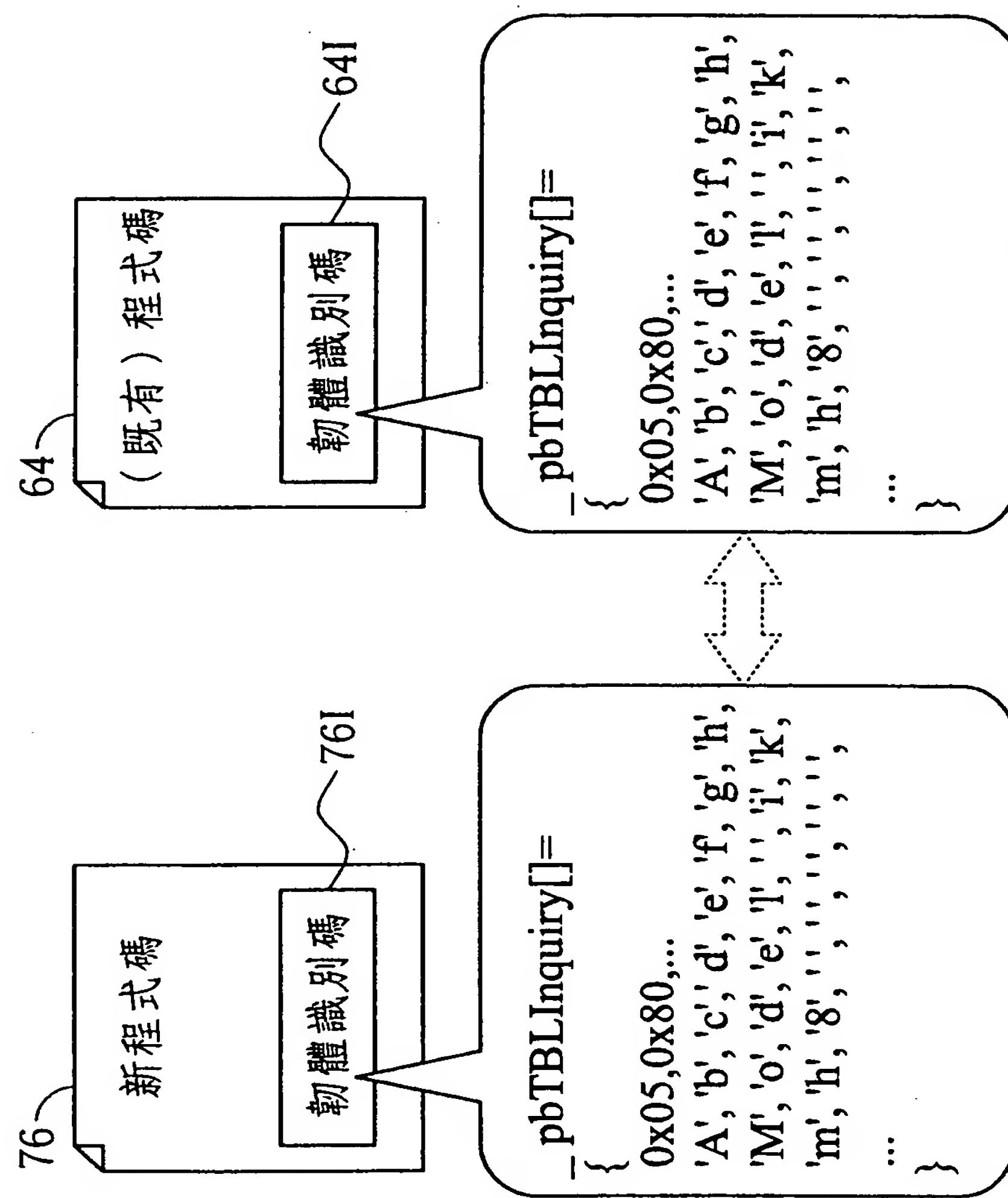


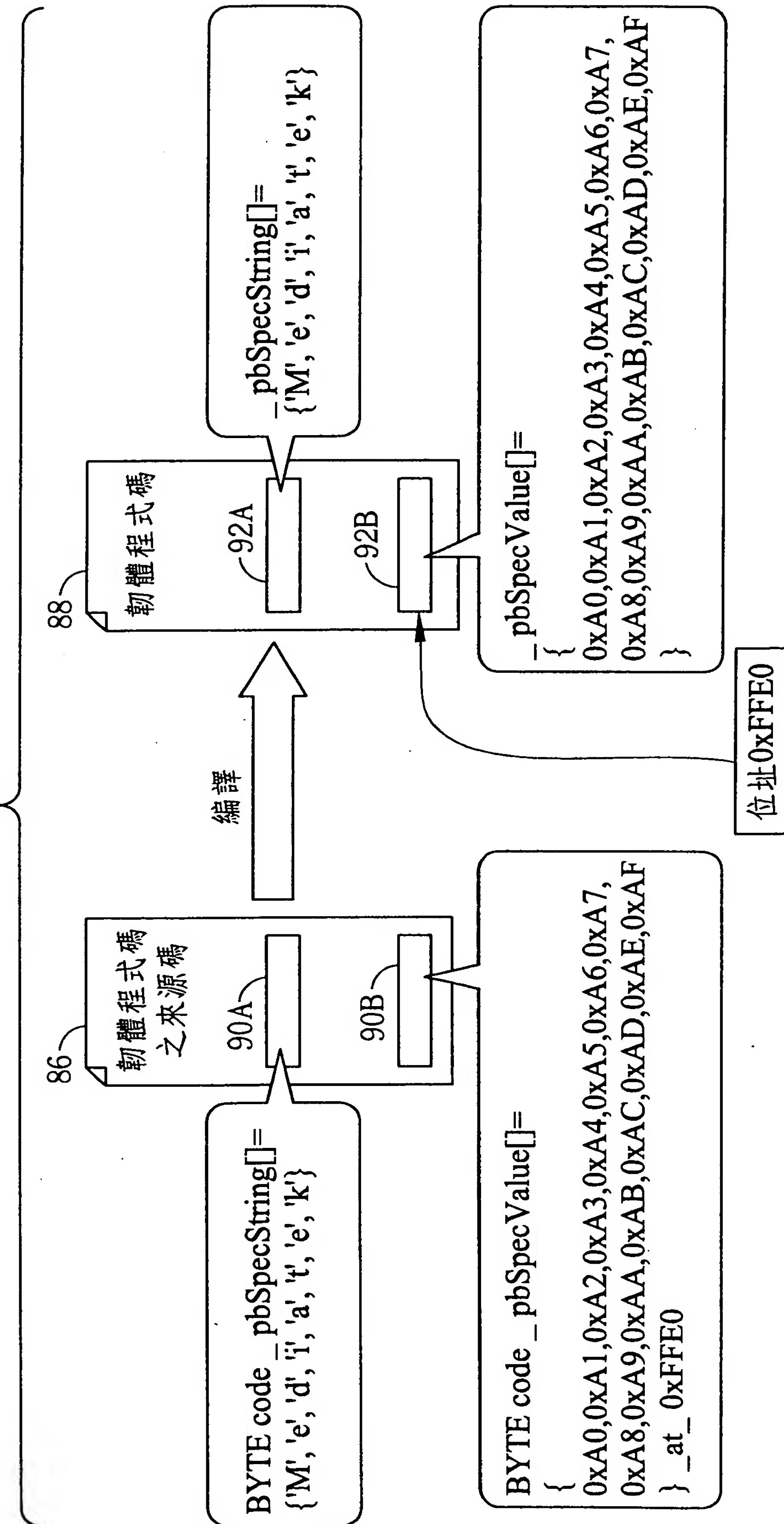
圖二



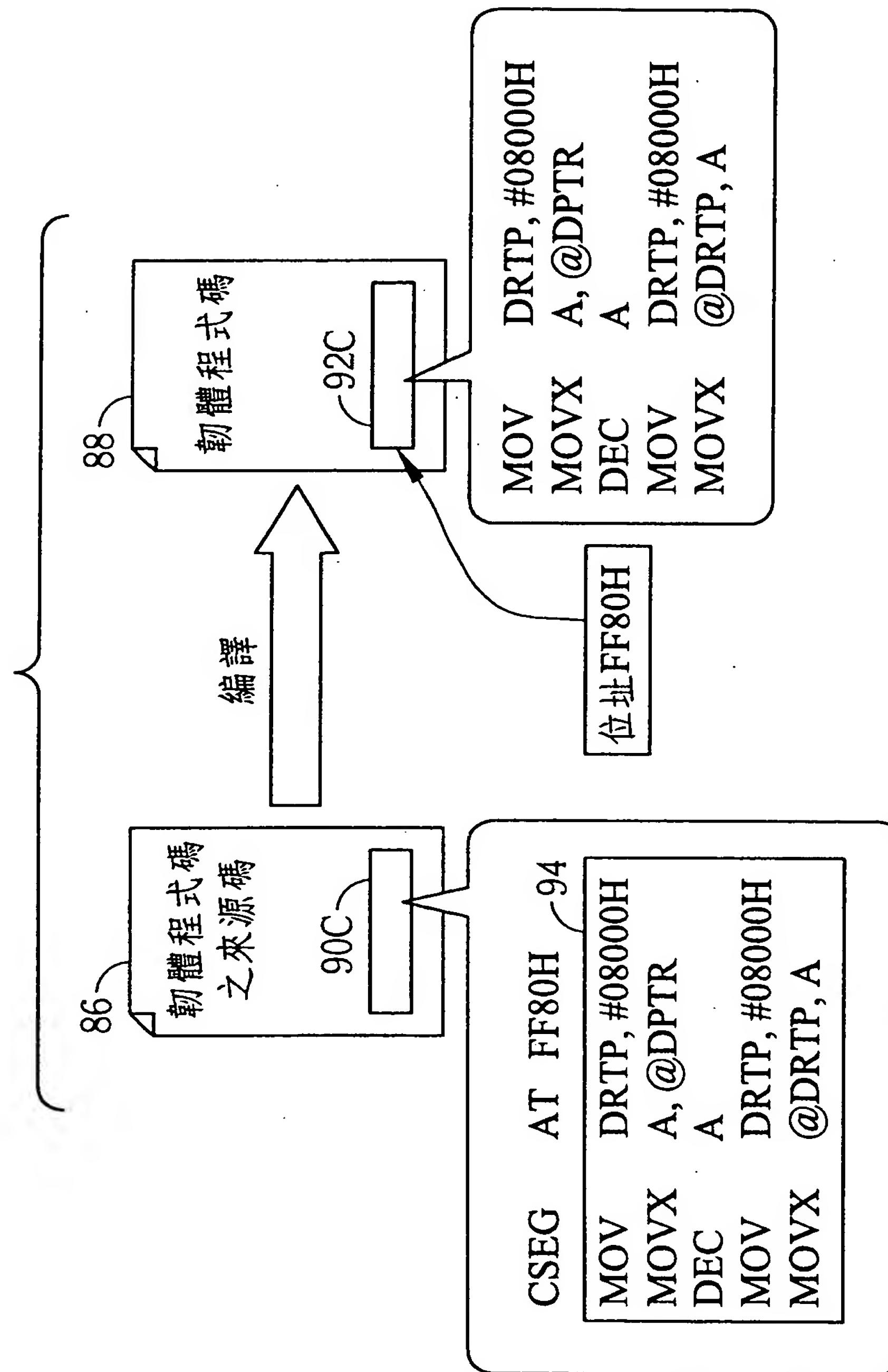
圖四

圖五 A

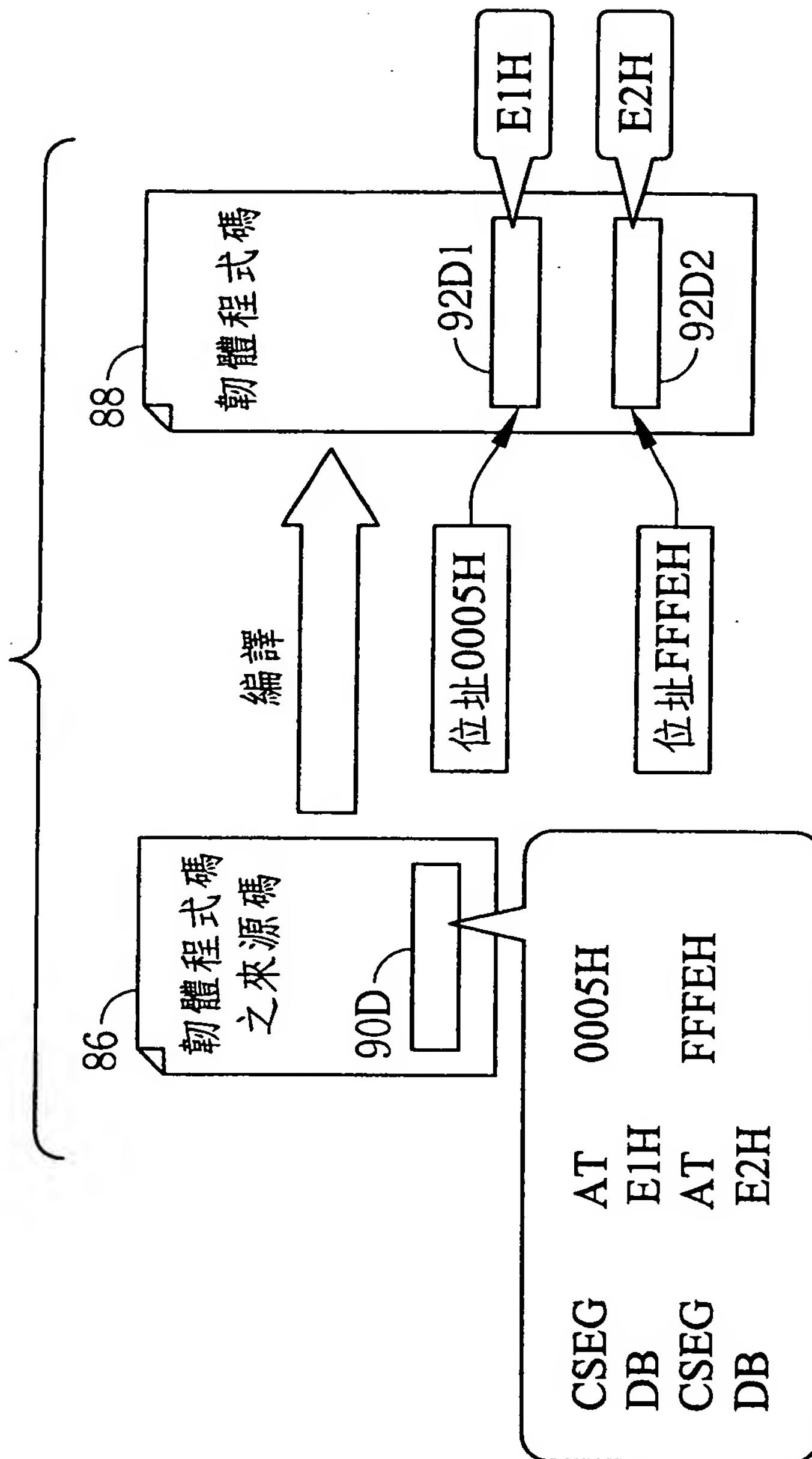




圖五 B

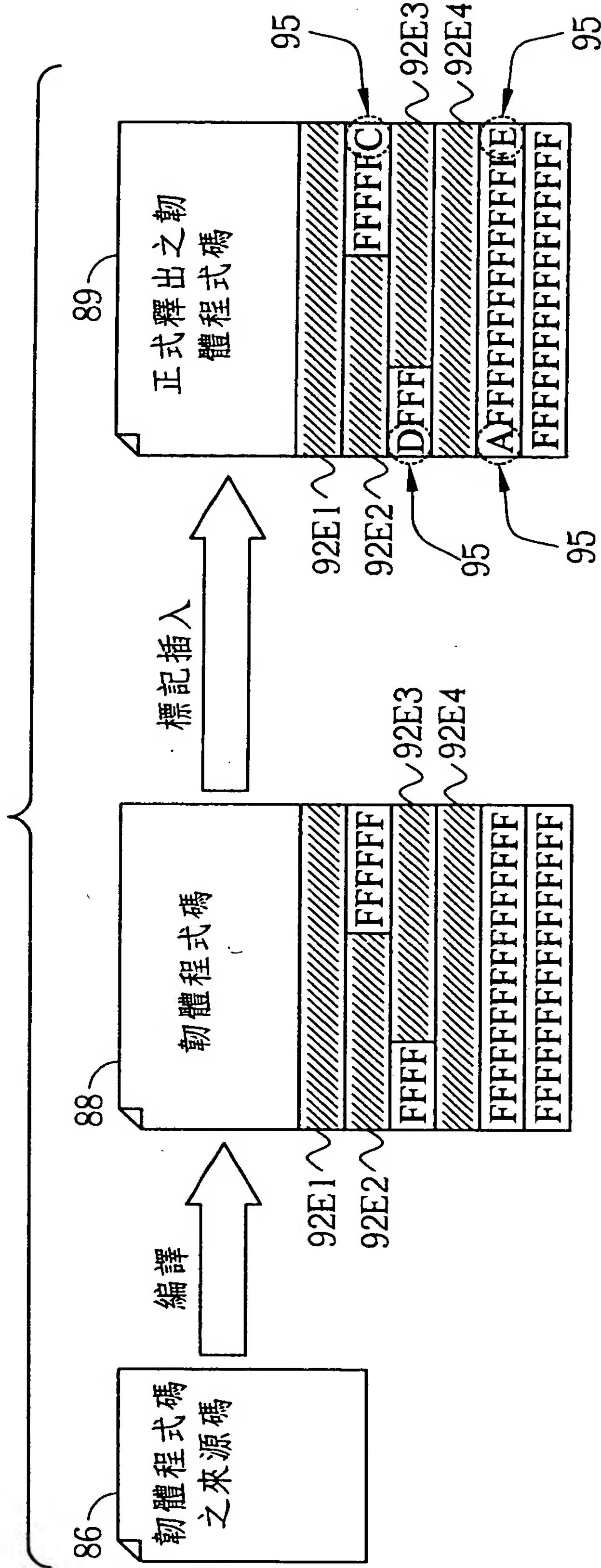


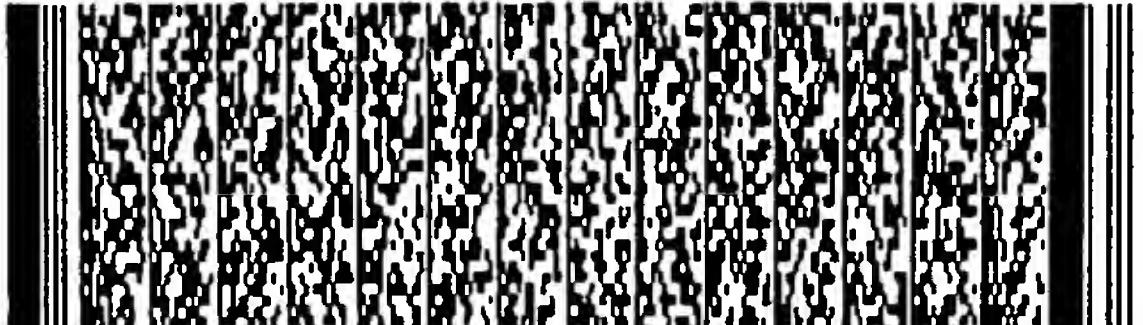
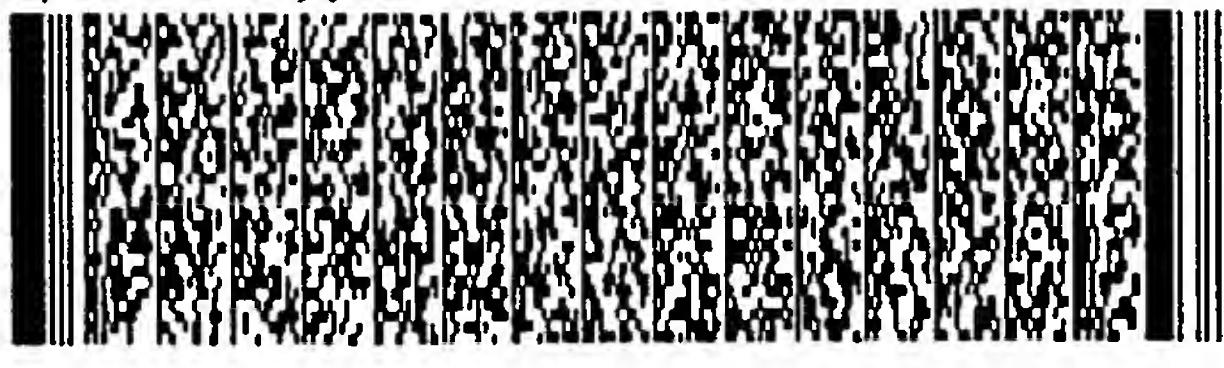
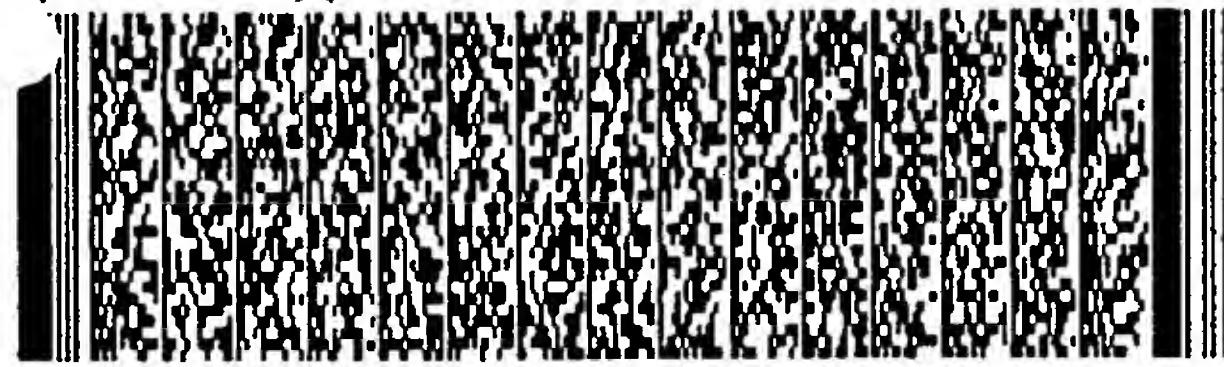
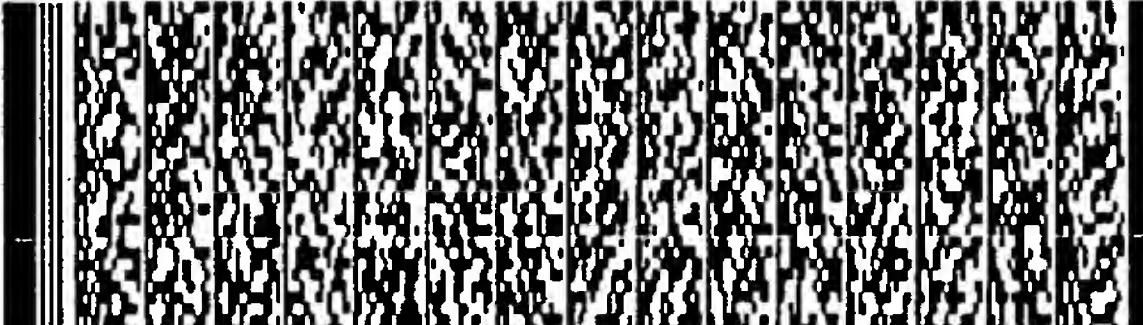
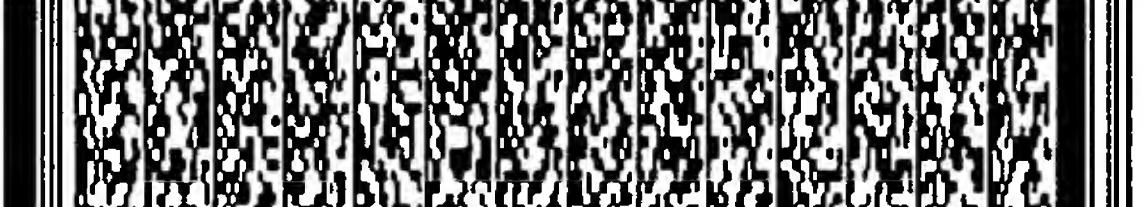
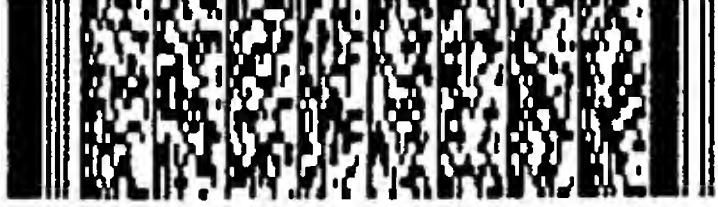
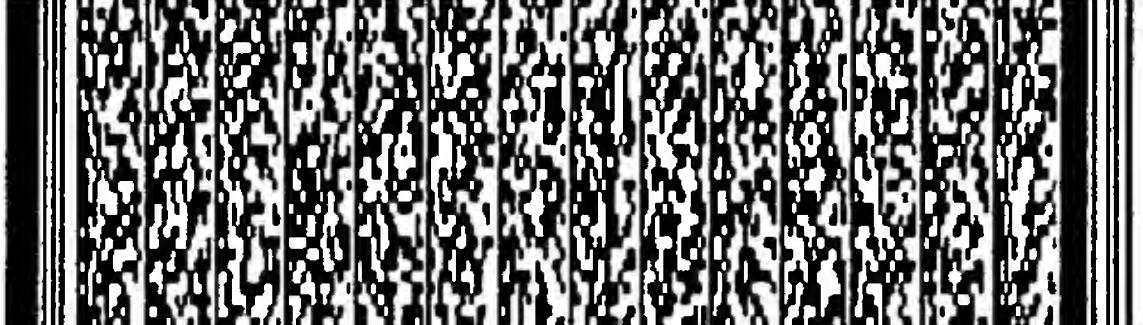
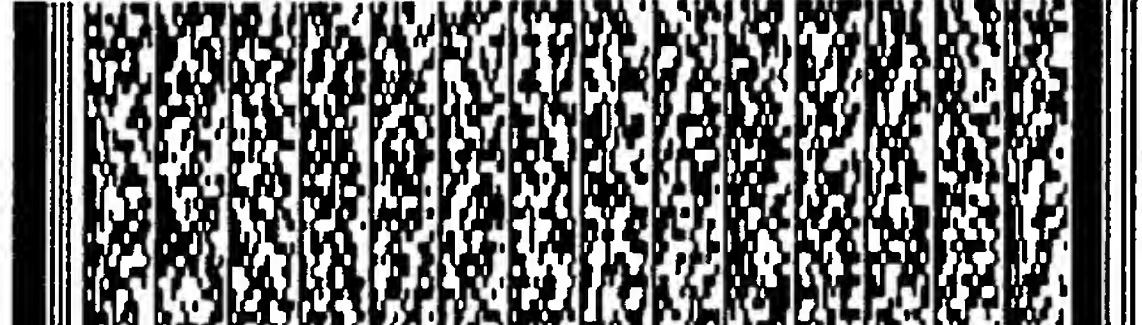
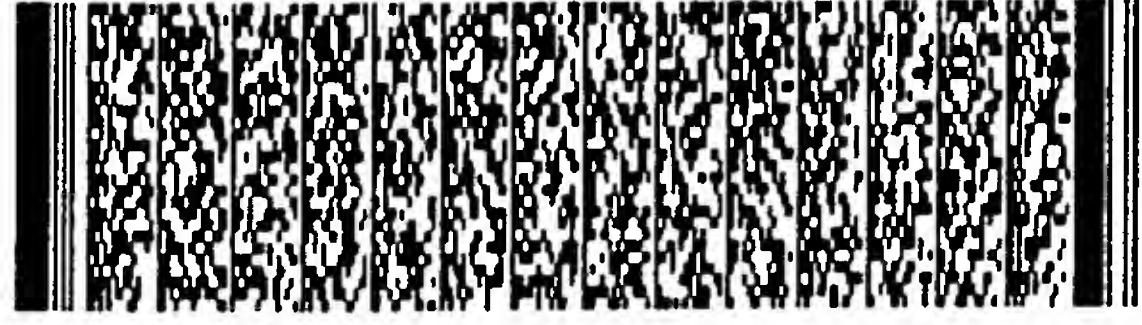
圖五 C



圖五 D

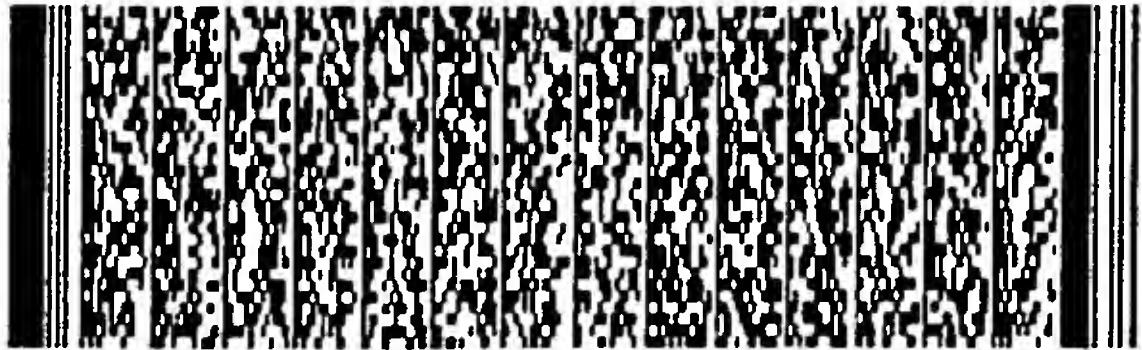
圖五 E



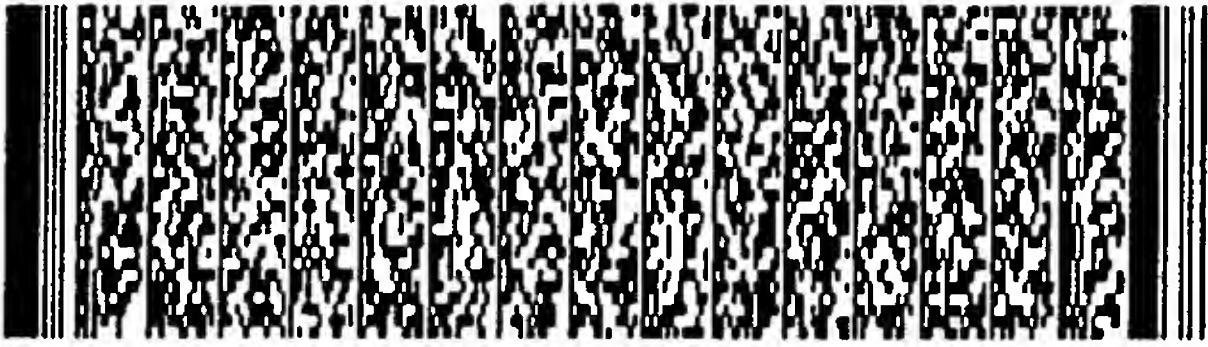


及相關裝置

第 10/52 頁



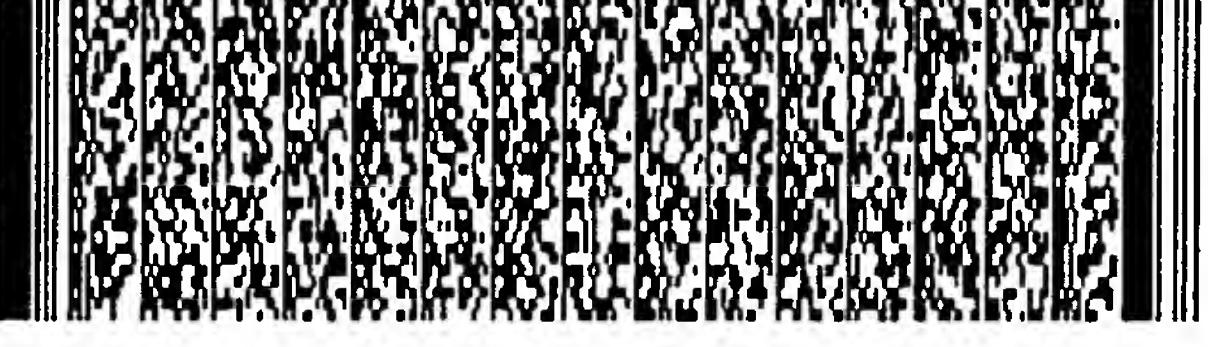
第 11/52 頁



第 12/52 頁



第 13/52 頁



第 14/52 頁



第 15/52 頁



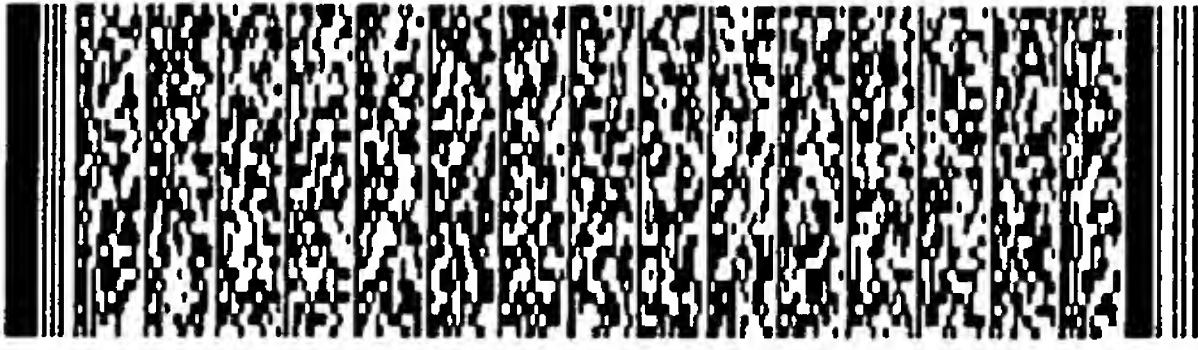
第 16/52 頁



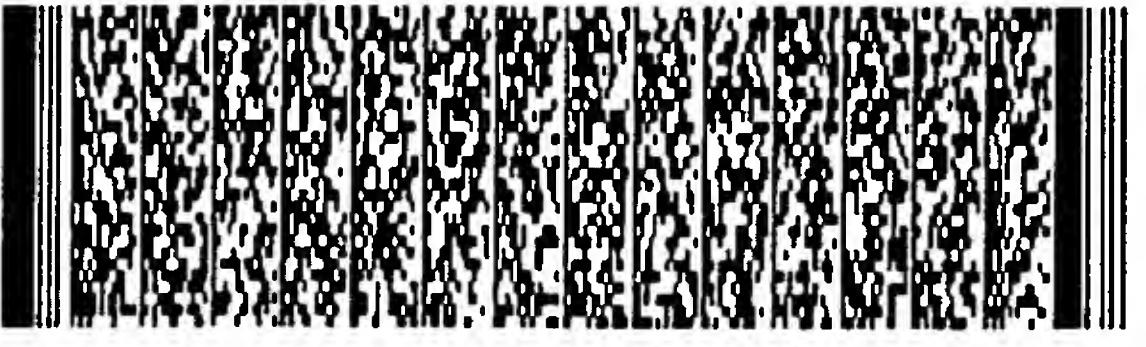
第 17/52 頁



第 11/52 頁



第 12/52 頁



第 13/52 頁



第 14/52 頁



第 15/52 頁



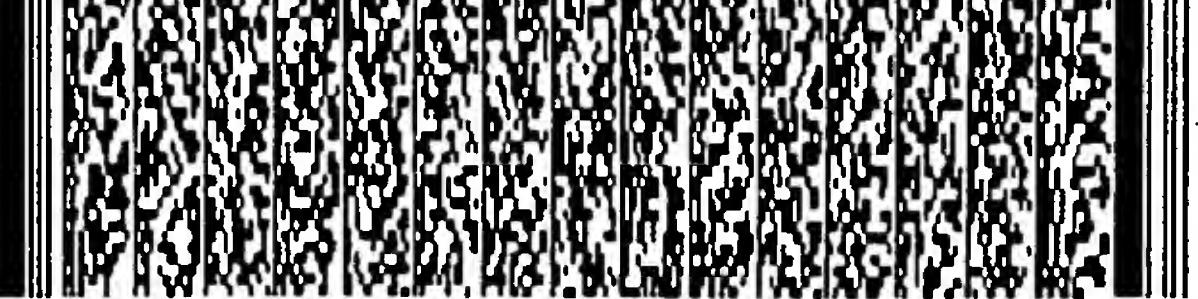
第 16/52 頁



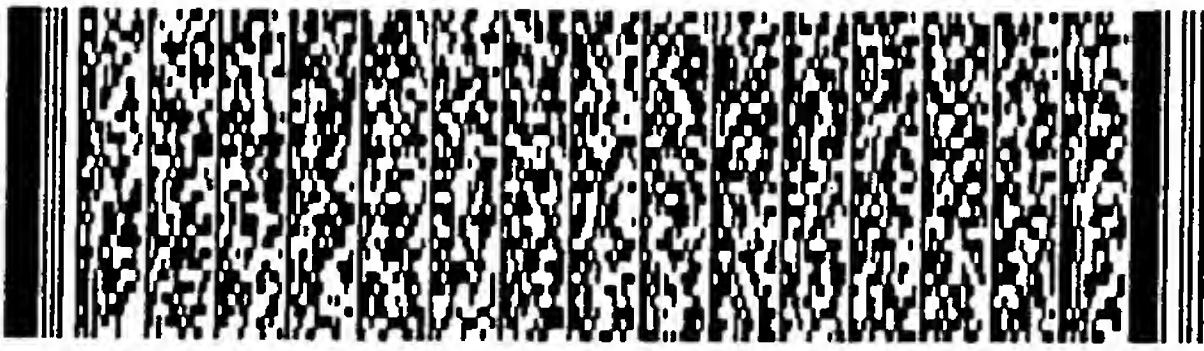
第 17/52 頁



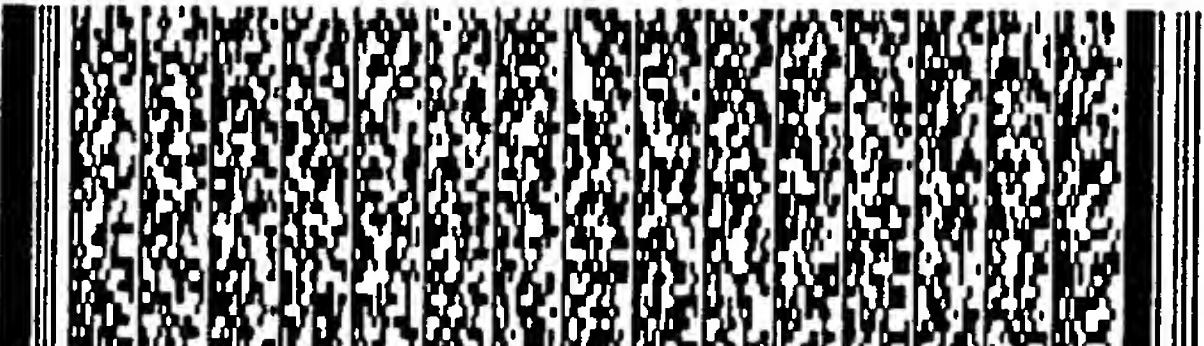
第 18/52 頁



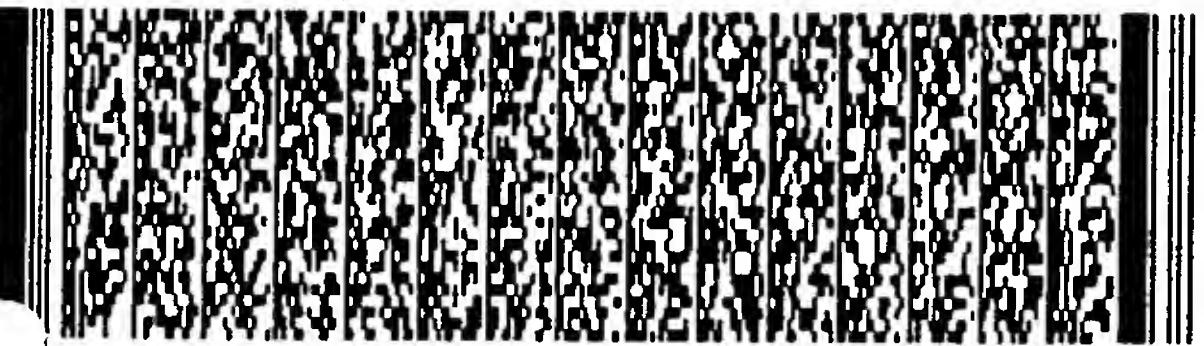
第 18/52 頁



第 19/52 頁



第 20/52 頁



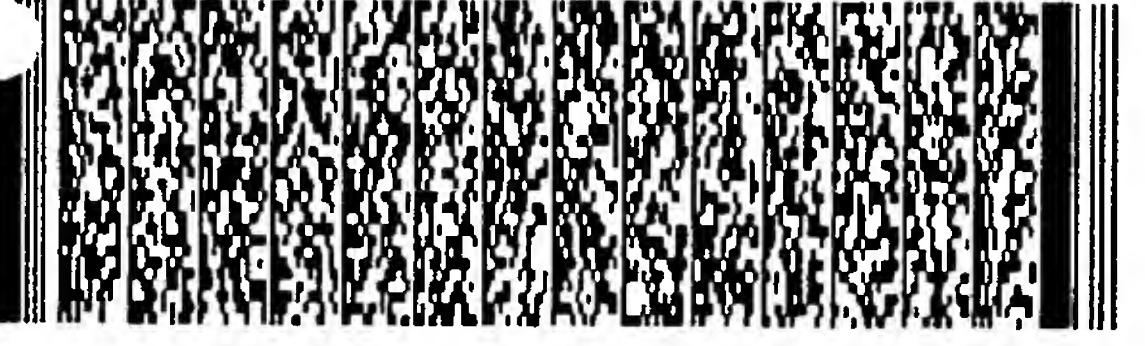
第 21/52 頁



第 22/52 頁



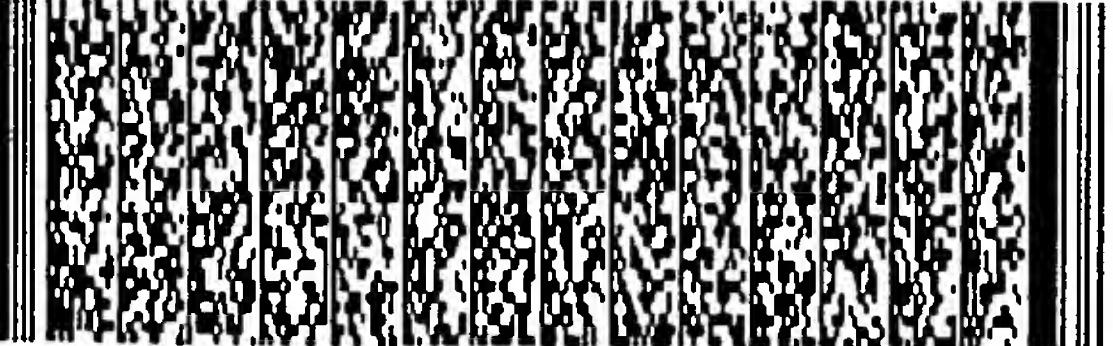
第 23/52 頁



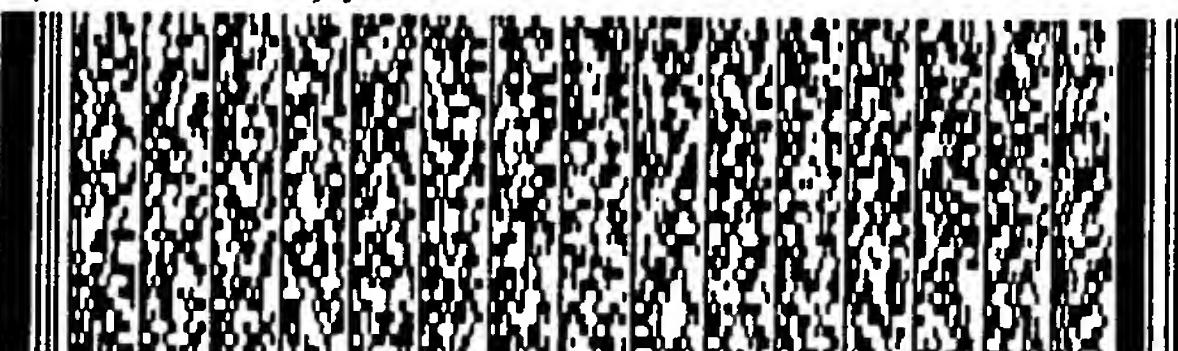
第 24/52 頁



第 25/52 頁



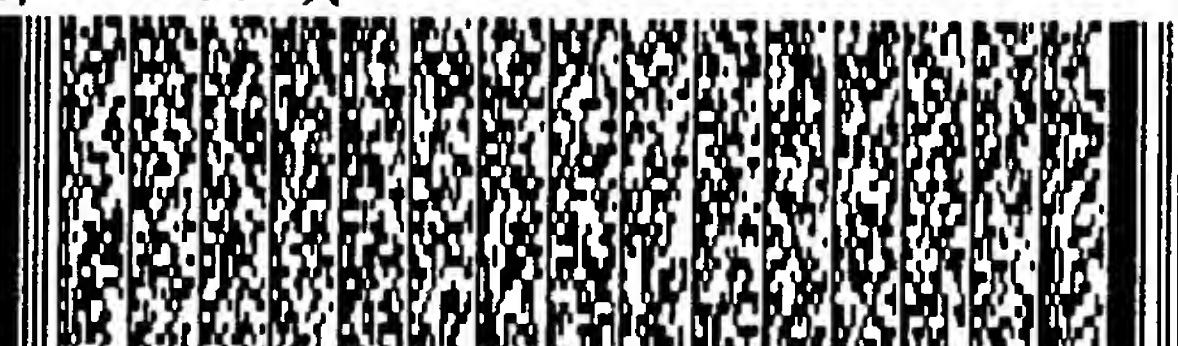
第 19/52 頁



第 20/52 頁



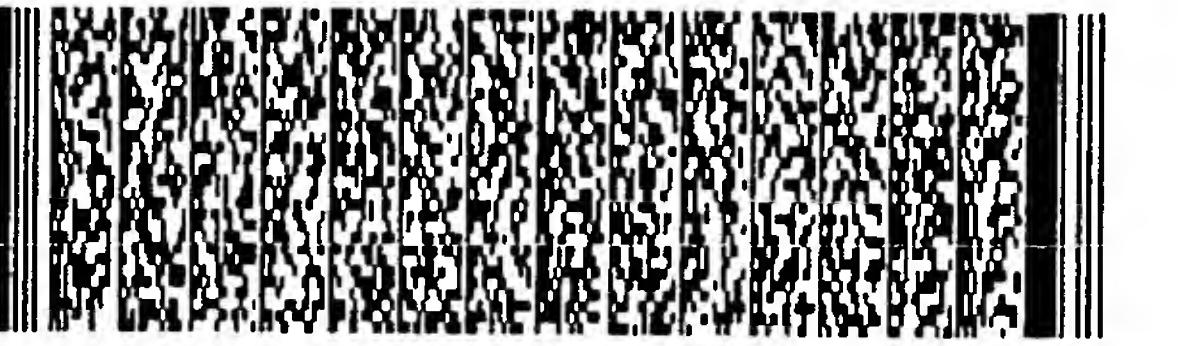
第 21/52 頁



第 22/52 頁



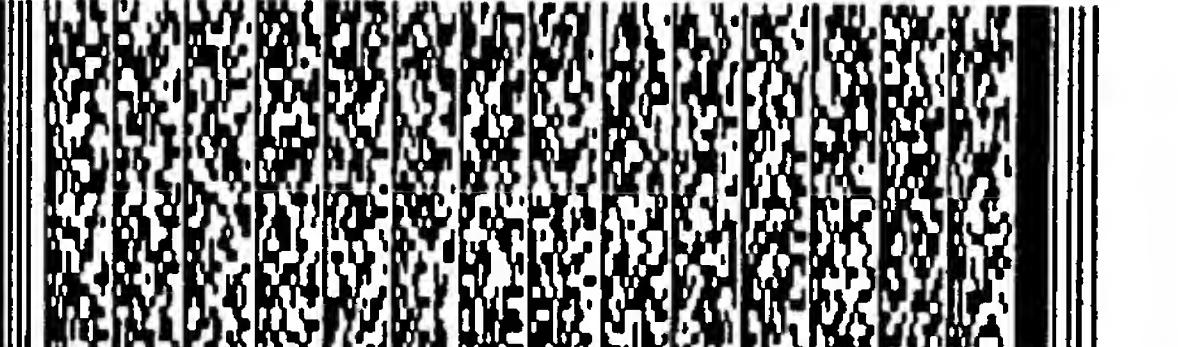
第 23/52 頁



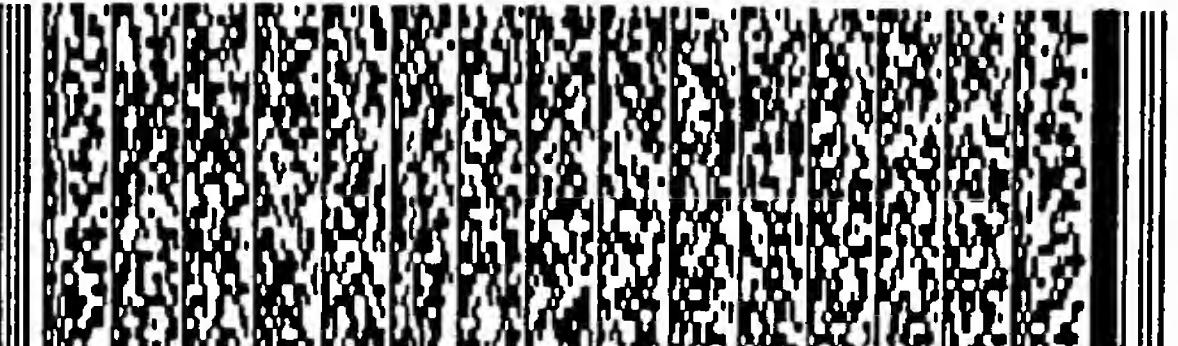
第 24/52 頁



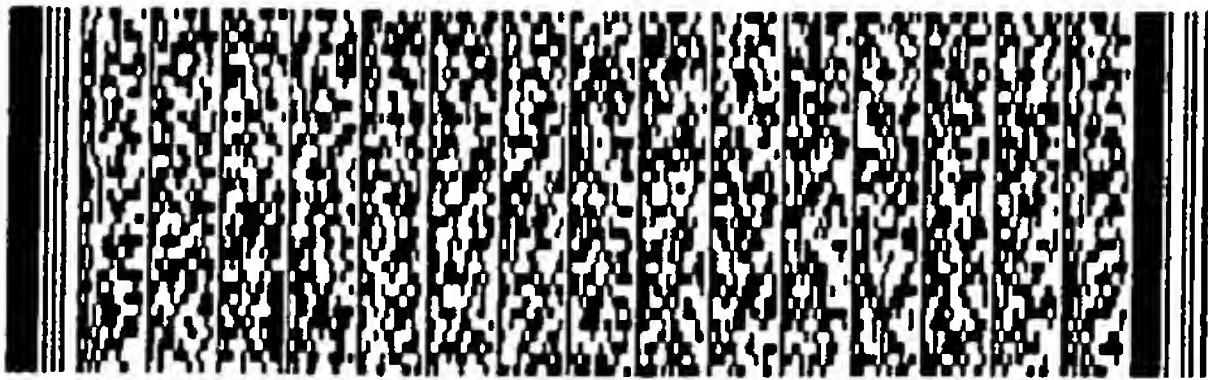
第 25/52 頁



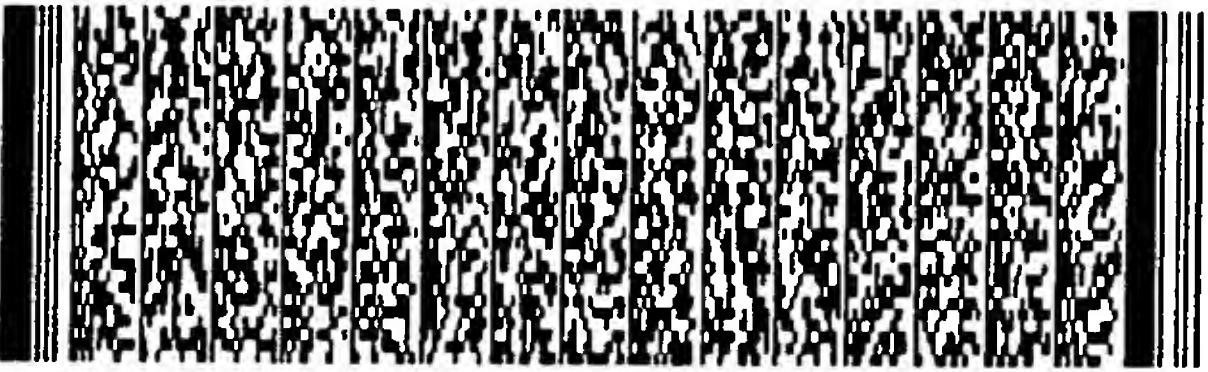
第 26/52 頁



第 26/52 頁



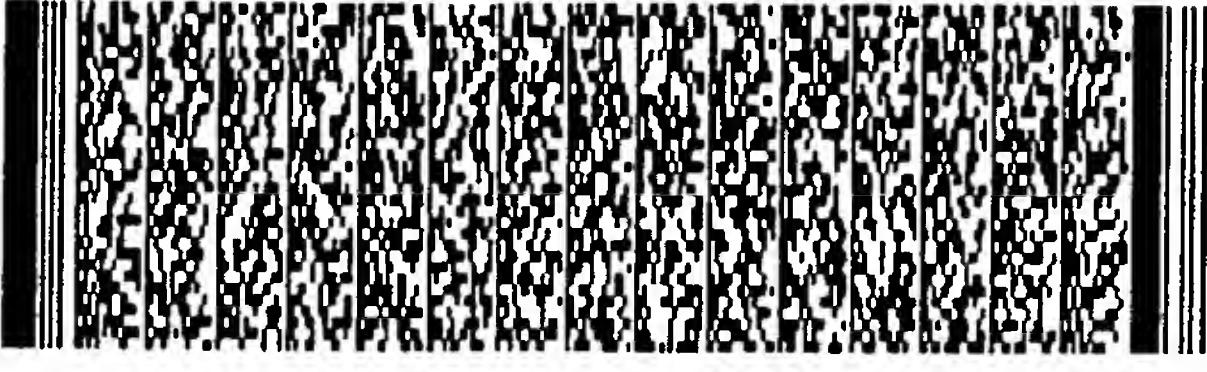
第 27/52 頁



第 28/52 頁



第 29/52 頁



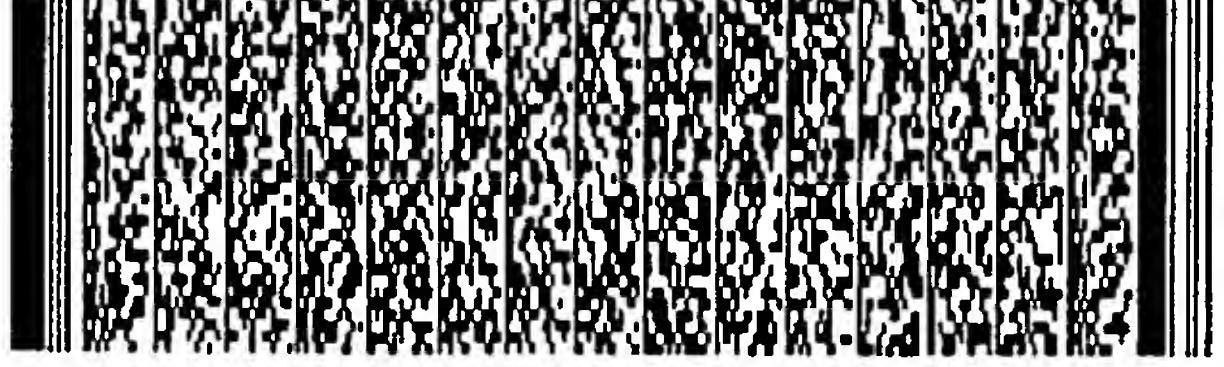
第 30/52 頁



第 31/52 頁



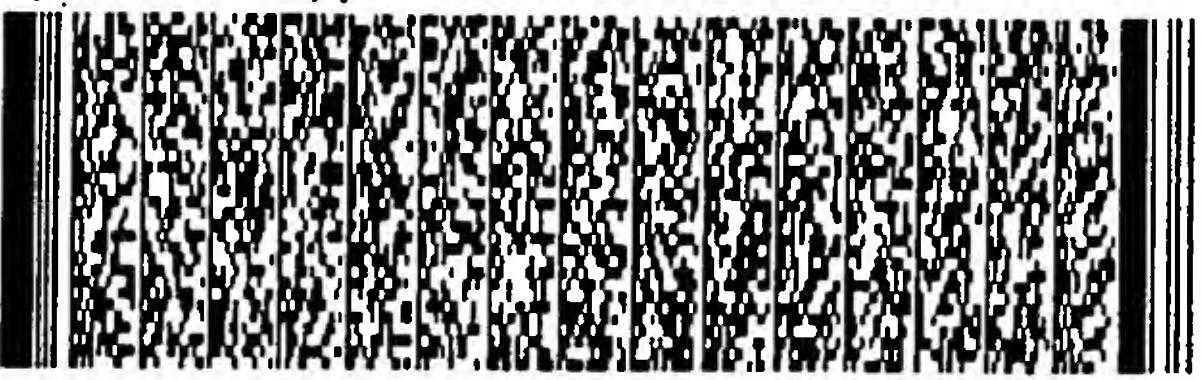
第 32/52 頁



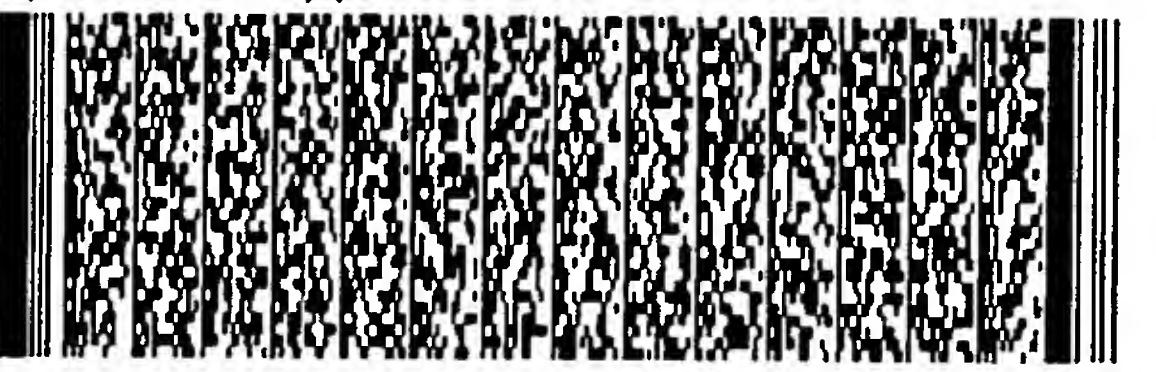
第 33/52 頁



第 27/52 頁



第 28/52 頁



第 29/52 頁



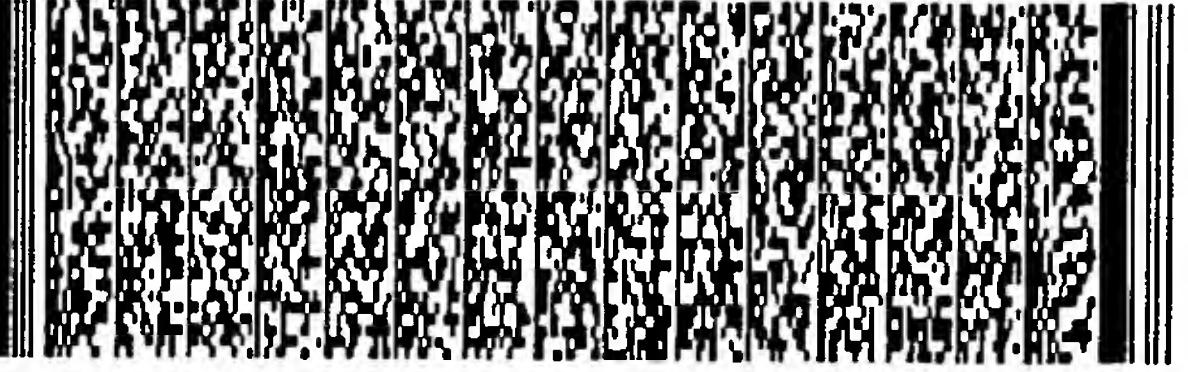
第 30/52 頁



第 31/52 頁



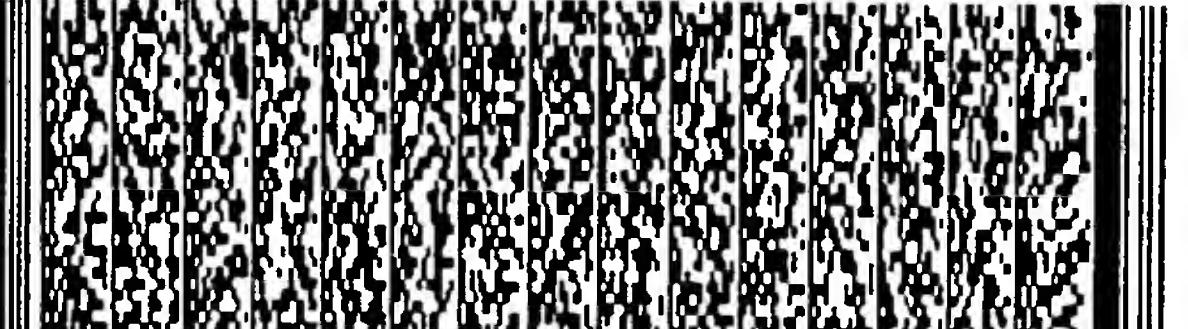
第 32/52 頁



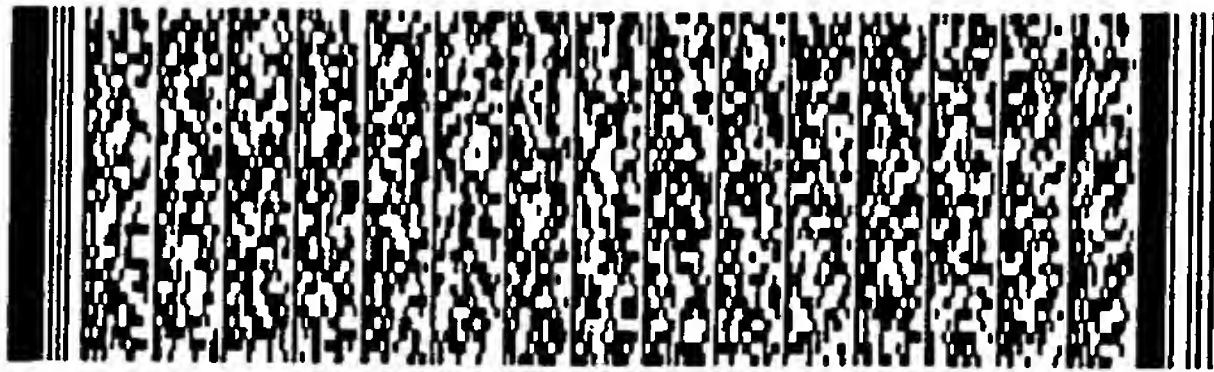
第 33/52 頁



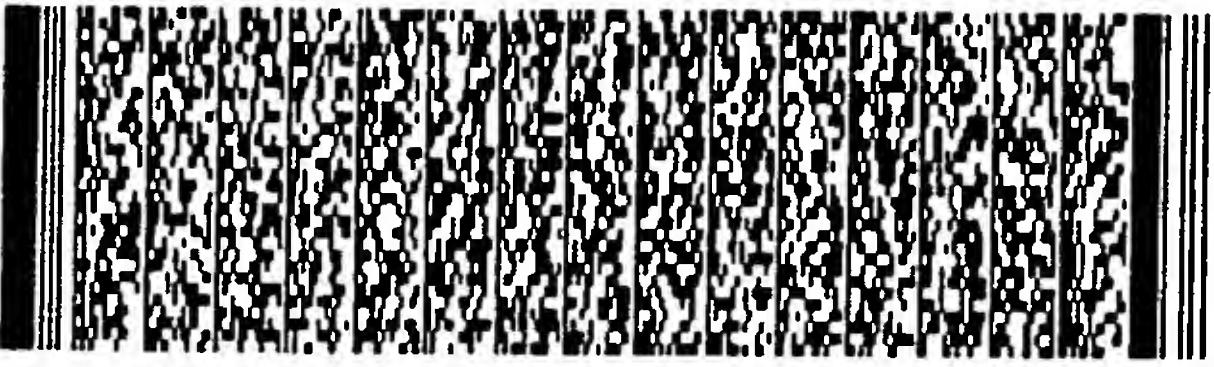
第 34/52 頁



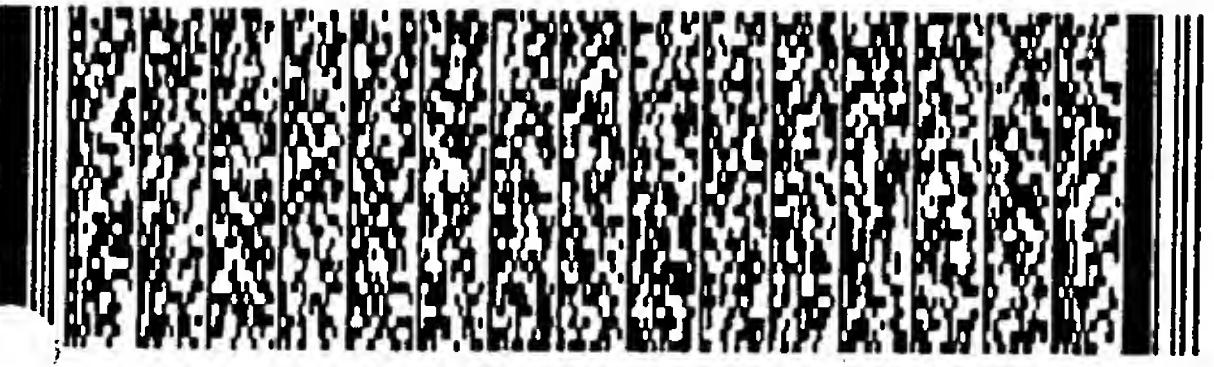
第 34/52 頁



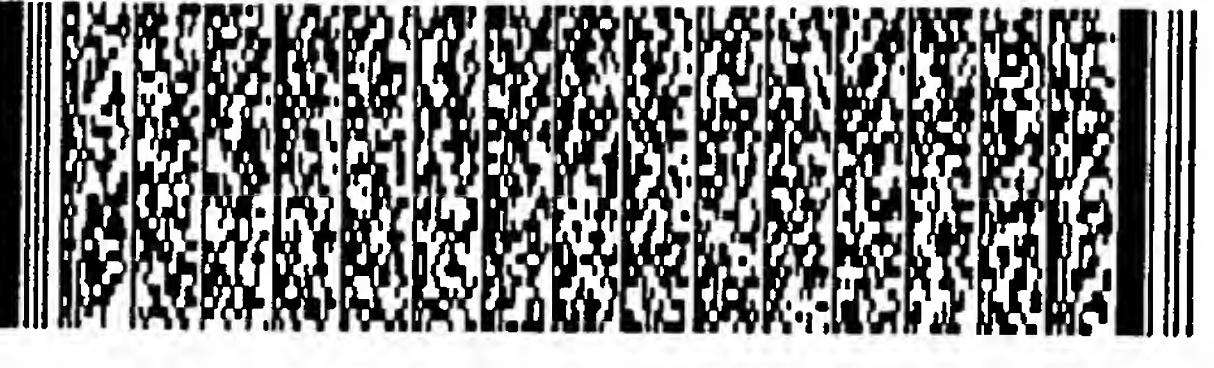
第 35/52 頁



第 36/52 頁



第 37/52 頁



第 38/52 頁



第 39/52 頁



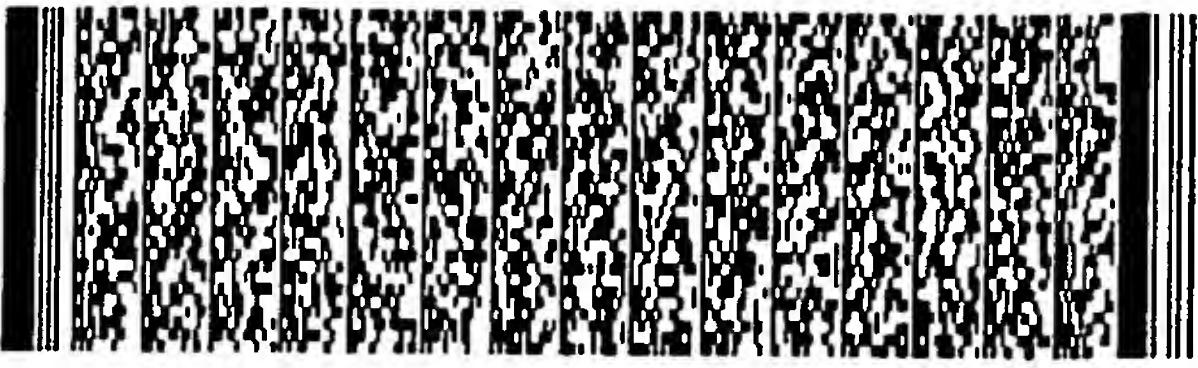
第 40/52 頁



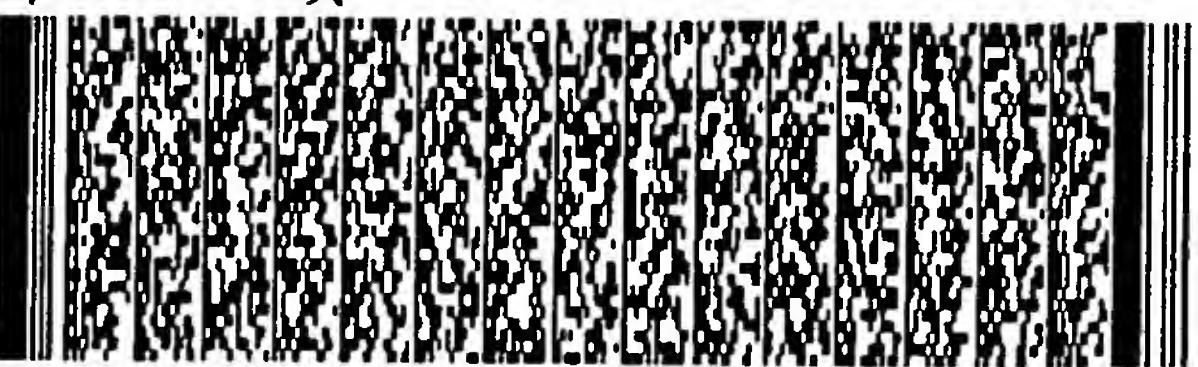
第 41/52 頁



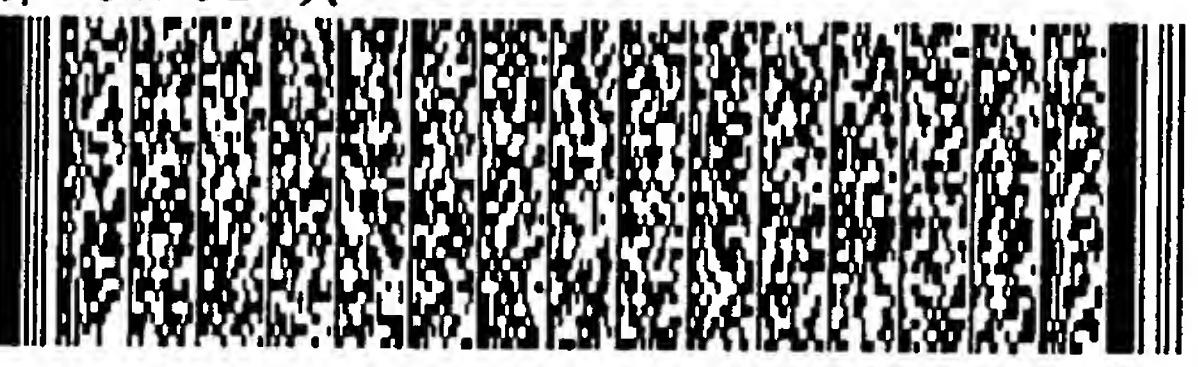
第 35/52 頁



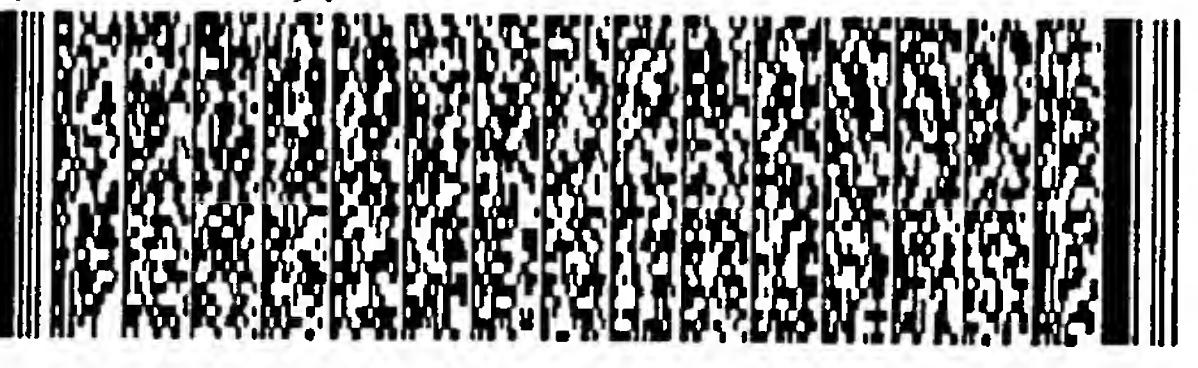
第 36/52 頁



第 37/52 頁



第 38/52 頁



第 39/52 頁



第 40/52 頁



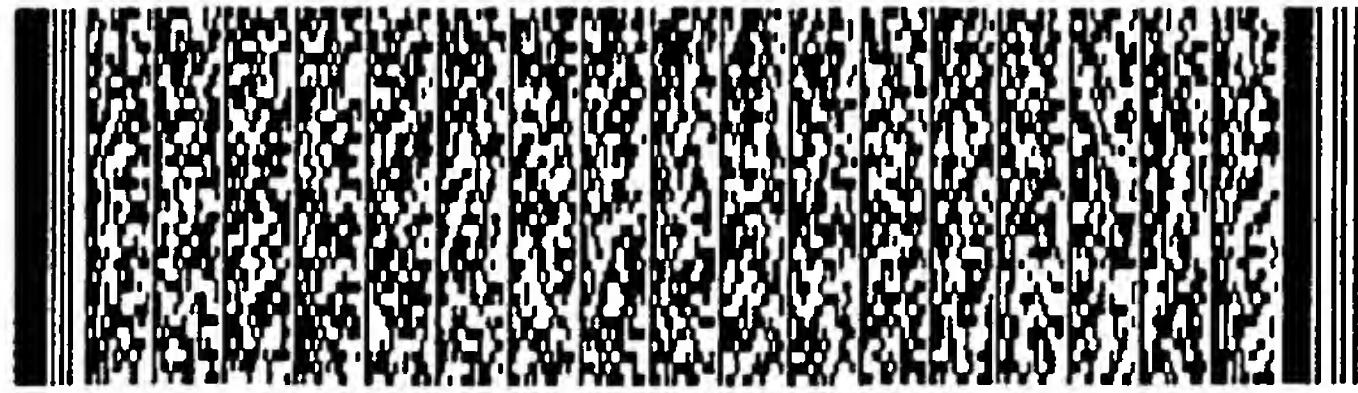
第 41/52 頁



第 42/52 頁



第 44/52 頁



第 46/52 頁



第 48/52 頁



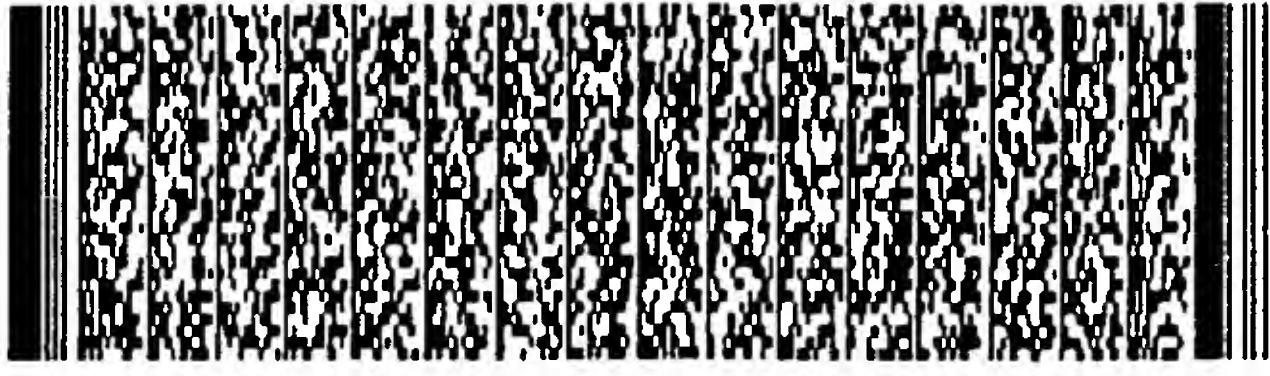
第 50/52 頁



第 51/52 頁



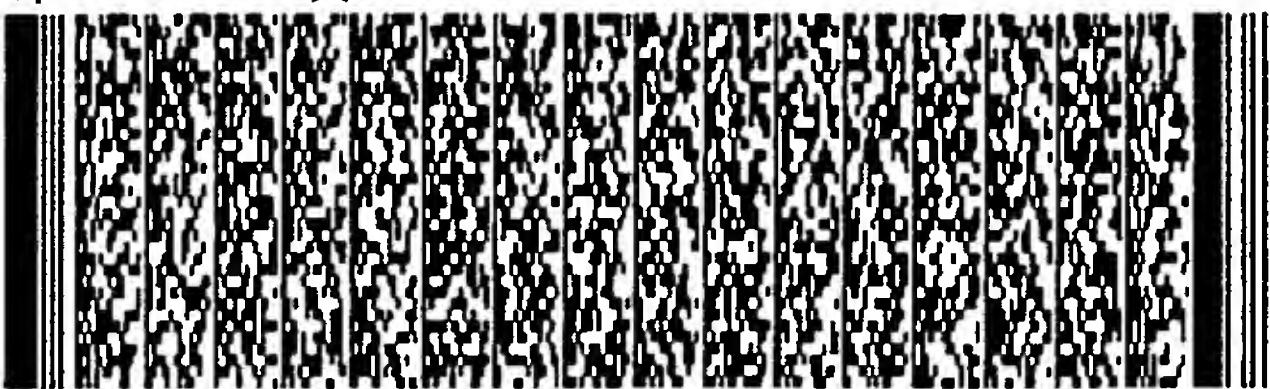
第 45/52 頁



第 47/52 頁



第 49/52 頁



第 51/52 頁



第 52/52 頁

